# המדריך השלם
# ASP 3

**Stephen Walther**

ייעוץ מקצועי:
מוטי סופר

SAMS

# המדריך השלם
# ASP 3

**הוראות התקנה והפעלה של התקליטור**
**בעמוד XXIII ובקובץ ONCD בתקליטור**

עורך ראשי: **זהר עמיהוד**

תרגום: **צור ריכטר-לוין** Microsoft Certified Professional, **יניב אברהם, יואב גולדברג**

עריכה מקצועית: **צור ריכטר-לוין** Microsoft Certified Professional, **מוטי סופר**

עריכה לשונית ועיצוב: **רמה שנקלר**

עיצוב עטיפה: **ישראל מצגר**



**לשם שטף הקריאה כתוב ספר זה בלשון זכר בלבד. ספר זה מיועד לגברים ונשים כאחד ואין בכוונתנו להפלות או לפגוע בציבור המשתמשים/ות.**

- **טלפון: 09-9564716**
- **פקס: 09-9571582**
- **דואר אלקטרוני: info@hod-ami.co.il**
- **אתר באינטרנט: www.hod-ami.co.il**

# המדריך השלם
# ASP 3

**Stephen Walther, et al.**

Designed for
Microsoft®
Windows NT®
Windows® 98

SAMS

# Active Server Pages Unleashed (Revised ASP 3.0)

By **Stephen Walther, et al.**

Editor: **Z. Amihud**

**הוצאת הוד-עמי**

**לספרי מחשבים בע"מ**

ת.ד. 6108 הרצליה 46160

טלפון: 09-9564716 פקס: 09-9571582

**info@hod-ami.co.il**



הודפס בישראל 2001



**HOD-AMI Ltd.**

P.O.B. 6108, Herzliya

ISRAEL, 2001

ISBN 965-361-272-7 מסת"ב

# תוכן עניינים מקוצר

# תוכן העניינים

## פרק 3: עבודה עם טפסי HTML ..... 62



## פרק 4: עבודה עם יישומי ASP ..... 84

# חלק 2: רכיבים ברי-התקנה של דפי ASP ........ 143

## פרק 7: יצירת תוכן דינמי מהשרת ............................ 145



## פרק 8: עבודה עם מערכת הקבצים ........................... 165

## פרק 9: מעקב אחר תצוגות הדף, והצגת פרסומות........ 185



## פרק 10: עבודה עם דואר אלקטרוני ......................... 205



## פרק 11:
## עבודה עם ממשקי שירות של Active Directory ........... 230

# חלק 3: שימוש בדפי ASP עם מסדי נתונים........ 269

## פרק 12: עבודה עם Connections ונתונים .................. 271

## פרק 13: עבודה עם Recordset .................................. 301



## פרק 14: עבודה עם האובייקט Command ................... 328



## פרק 15:
## עבודה עם Index Server וחיפוש בטקסט-מלא............ 348

## פרק 18: יצירת רכיבים באמצעות Visual Basic ....... 417

## פרק 22: שימוש ב-ASP עם MSMQ (Microsoft Message Queue Server) .......... 533

## חלק 6: נושאים מתקדמים ............................ 563

## פרק 26: Win 2000, IIS 5.0, ASP - כולם ביחד........... 625



# חלק 7: דוגמאות ליישומי ASP........................ 645

## פרק 27: יצירת פורום דיון....................................... 647

## פרק 28: יצירת חנות מקוונת.................................. 664

## חלק 8: נספחים - Appendixes

# התקליטור המצורף

## מה בתקליטור?

הוצאת הוד-עמי מגישה לך תקליטור הכולל:

- קטלוג HTML - קטלוג ספרי המחשבים האינטראקטיבי של הוצאת הוד-עמי. הקטלוג מאפשר קריאת פרקים לדוגמה, תוכן עניינים ועוד.

  לשם קריאת הפרקים לדוגמה יש להתקין את תוכנת Adobe Acrobat Reader אשר מצורפת בתקליטור. הוראות התקנה בהמשך. הקטלוג מומלץ לצפייה באמצעות Internet Explorer גירסה 5 ומעלה. הדפדפן מצורף בתקליטור. הוראות התקנה בהמשך. התקנת שתי התוכנות קלה וניתנת לביצוע באמצעות קישור ישירות מהקטלוג.

- תוכנות עזר שימושיות.

---

**הערה**: אם מנהל התקן כונן התקליטורים המותקן הוא 16 סיביות - ייתכן שתראה רק 8 תווים ראשונים של שם קובץ (במידה שהמקור ארוך יותר).

**הסיבה**: כונני תקליטורים במהירות x4 עובדים עם מנהל התקן שעבד בסביבת DOS ו-Windows 3.11 ויכול לעבוד גם עם Windows 95, למעט היכולת לזהות קבצים עם שמות ארוכים.

**הפתרון:** להתקין מנהל התקן 32 סיביות (אם קיים), או לקנות כונן תקליטורים חדש ולוודא שמצורף אליו מנהל התקן 32 סיביות.

---

**אם אינך בטוח כיצד להתקין את התוכניות שבתקליטור פנה לאיש מקצוע.**

**קרא את קובץ ONCD שבתקליטור כדי לקבל עוד מידע לגבי התקליטור**

**התיקיה הרלוונטית לספר זה - Books\59303**

# העתקת הקבצים הרלוונטיים לספר מהתקליטור

קבצי התוכניות נמצאים בתיקיה נפרדת - **התיקיה Books\59303** אשר בתקליטור.

כדי להעתיק את תכולת התיקיה הרלוונטית לספר זה:

1. לחץ על לחצן **התחל**, **תוכניות**, **סייר Windows**.
2. לחץ על כונן התקליטורים וסמן את התיקיה הרלוונטית שבתיקיה Books.
3. גרור אותה לכונן הדיסק הקשיח.

מכיון שמקור הקבצים הוא התקליטור, הם מסומנים לקריאה בלבד. יש לשנות מאפיין זה כך:

1. ב**סייר Windows** היכנס לתיקיה בדיסק הקשיח בה נמצאים הקבצים שהעתקת.
2. סמן את כולם על ידי **Ctrl+A**.
3. הצב את סמן העכבר מעל האזור המסומן ולחץ לחיצה ימנית בעכבר.
4. מתפריט הקיצור בחר **מאפיינים**.
5. בטל את הסימון בתיבה **קריאה בלבד** (דאג שהיא תהיה ריקה).
6. לחץ על **החל**, לחץ **אישור**.

**הערה**: יש לסמן את הקבצים בכל תיקיה בנפרד.
בדוק האם השתנה המאפיין על ידי סימון הקובץ: לחיצה ימנית על הקובץ ומתפריט הקיצור בחירה באפשרות **מאפיינים**. שים לב שתיבת הסימון **קריאה בלבד** תהיה ריקה.

# Acrobat Reader - התקנה

יש להתקין תוכנה זו כדי לקרוא ולהדפיס את הפרקים לדוגמה, אליהם ניתן לגשת באמצעות **קטלוג HTML** (שהתקנתו תוסבר בהמשך). התוכנה גם מאפשרת חיפוש בעברית ובאנגלית במסמך המוצג. בנוסף, בעזרת תוכנה זו תוכל לקרוא את המסמכים שההוצאה מפרסמת באתר האינטרנט. התוכנה פועלת במערכות הפעלה **Windows 95 ומעלה.**

1. לחץ על לחצן **התחל** ובחר באפשרות **הפעלה**.
2. בתיבת הטקסט הקלד את הפקודה:
**X:\Software\Adobe Acrobat\ARME4Heb.exe** (החלף את האות X באות המייצגת את כונן התקליטורים שלך) ולחץ על **אישור**.
3. אשף ההתקנה מתקין את הרכיבים הנדרשים. עליך ללחוץ על **Next**, **Accept** ו-**Next** פעם נוספת כדי לבצע את ההתקנה.

**4.** בסיום ההתקנה עשויה להופיע על המסך תיבת דו-שיח **התנגשות בין גירסאות** ומייד אחר כך להיעלם. במקומה תופיע על המסך תיבת הודעה של תוכנית ההתקנה. לחץ **אישור** ובתיבת הדו-שיח **התנגשות בין גירסאות** ששבה להופיע לחץ **כן**, כדי לשמור את גרסת הקובץ שלך.

# קטלוג HTML

הוצאת הוד-עמי גאה לבשר על קטלוג HTML העושה שימוש בטכנולוגיות אינטרנט מתקדמות כדי להביא לך את המידע על ספרי המחשבים המקצועיים שלנו בלחיצת לחצן העכבר.

התקנת הקטלוג אינה מעתיקה קבצים לכונן הדיסק הקשיח המקומי, אלא רק מציבה סמל קיצור דרך אל הקטלוג על שולחן העבודה שלך. להפעלת הקטלוג חייב התקליטור להימצא בכונן. הקטלוג מומלץ לצפייה באמצעות **Microsoft Internet Explorer**.

בעזרת קטלוג HTML תוכל:

**1.** לעיין במידע אודות ספרי ההוצאה מתי שתרצה (לחיצה כפולה.... וזהו!).

**2.** לעבור במהירות ובקלות בין הקטלוג והיישום בו אתה עובד.

**3.** לעיין במידע אודות כל ספר וספר.

**4.** לצפות ואף להדפיס פרק לדוגמה.

**5.** לצפות ואף להדפיס מגה-אינדקס של הספר.

**6.** לגשת במהירות, בגישה אינטואיטיבית, תוך התמקדות מהירה בספר המבוקש.

**7.** לעיין בקטלוג בקצב אישי שלך.

**8.** לנווט את דרכך בקטלוג ולחזור ולהתרענן בכל נושא בכל רגע.

**הקטלוג ניתן לצפייה באמצעות Internet Explorer**

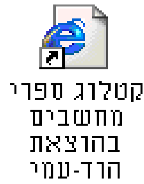
**1.** הכנס את התקליטור לכונן.

**2.** לחץ על **התחל** ובחר **הפעלה**.

**3.** בעזרת לחצן **עיון** סמן את הקובץ **Setup.EXE** אשר בתיקיה הראשית של התקליטור המצורף.

**4.** לחץ **פתח**.

**5.** לחץ **אישור**.




1. ודא שתקליטור הוד-עמי נמצא בכונן התקליטורים.
2. הפעל את הסמל עם הכיתוב **קטלוג ספרי מחשבים בהוצאת הוד-עמי** שעל שולחן העבודה.

# מה עוד בתקליטור?

הוצאת הוד-עמי מפיצה תוכנות אלו כבונוס ללקוחות ההוצאה, ואינה מתיימרת לגבות תשלום עבור התוכניות המצורפות, ו/או לתמוך בהם.

---

**אזהרה**: השימוש בתוכן תקליטור זה הוא על אחריותו הבלעדית של המשתמש. המוצרים המותקנים בתקליטור זה מסופקים באחריות החברות המייצרות אותם, בהתאם לתנאי האחריות המצורפים לכל אחד מהמוצרים. הוצאת הוד-עמי אינה אחראית, בכל צורה שהיא, לאופן ולטיב התוכנות המותקנות.

---

בכל שאלה לגבי תוכנה הנמצאת בתקליטור, יש לפנות למפתחי התוכנה (כל תוכנה בנפרד), כפי שמצוין בקבצי העזרה של התוכנה המדוברת.

הקבצים הם גרסאות שיתופיות (ShareWare) וחופשיות (FreeWare).

גירסה שיתופית (ShareWare) מאפשרת לך, המשתמש, לבדוק את יעילות התוכנה ואת תאימותה לעבודה אותה אתה מבצע. אם התוכנה נמצאה מתאימה לצרכיך, עליך לשלם למפתחיה תשלום סמלי (לפי הרשום בקבצי העזרה של כל תוכנה ותוכנה בנפרד) כדי לקבל רישיון מלא לשימוש בה. רכישת רישיון לשימוש בתוכנה יפתח בפניך מיגוון אפשרויות שייתכן ולא עמדו לרשותך בהפעלת הגירסה השיתופית.

# התקנת תוכנת גלישה לאינטרנט Microsoft Internet Explorer 5.5

תוכנית ההתקנה מזהה את גרסת מערכת ההפעלה ומתקינה את גרסת הדפדפן הדרושה.

ניתן להתקנה בכל מחשב בו מותקנת מערכת הפעלה Windows 95, Windows 98, Windows Me, Windows 2000 בממשק משתמש עברי או אנגלי.

1. הכנס את התקליטור לכונן.
2. לחץ על לחצן **התחל** ובחר באפשרות **הפעלה**.
3. לחץ על לחצן **עיון**.
4. בחר בכונן התקליטורים בתיקיה **Software\IE55** ובקובץ בשם **Setup.exe**.
5. לחץ על לחצן **פתח**.
6. לחץ על לחצן **אישור**.
7. פעל לפי ההוראות על המסך.

---

**אזהרה**: לפני ביצוע שדרוג מ- Windows 95 ל- Windows 98 בעברית (זו בה התפריטים בעברית ולחצן התחל מימין שורת המשימות), יש להסיר כל התקנה קיימת של Internet Explorer 5. לאחר ביצוע השדרוג ניתן לבצע התקנה מחדש, של הגירסה המתאימה.

---

# תיקיה ראשית SoftWare (הרשימה חלקית ועשויה להשתנות)

**הערה**: תוכנות להן יש גירסה מיוחדת עבור Windows 2000 יסומנו בתיקיה ששמה מסתיים ב- 2k (למשל, בתיקיה ICQ נמצא קובץ התקנה לתוכנה זו המתאים לכל גירסאות Windows, **ו**בתיקיה ICQ2k נמצא קובץ התקנה עבור מערכת ההפעלה Windows 2000 **בלבד**).

בדרך כלל לחיצה כפולה על שם הקובץ המפורט ברשימה מפעילה את תוכנית ההתקנה.

| שם תוכנה | תיאור | קובץ הפעלה | מבצע |
|---|---|---|---|
| Adobe Acrobat | תוכנה לצפייה בקבצי pdf, כולל תמיכה בעברית | ARME4Heb.exe | התקנה |
| FontsPekan | גופנים בעברית | FontsPekan.exe | פריסה |
| Power Toys | תוכניות שירות עבור Windows 95 | W9xPowerToys.exe | פריסה |
| Power Toys | תוכניות שירות עבור Internet Explorer | IEPowerToys.exe | פריסה |
| WinZip | תוכנית לפריסה/דחיסה של קבצים | WinZip80.exe | התקנה |

**פנה לקובץ ONCD לקבלת מידע על תוכנות נוספות.**

# הקדמה

## מבוא

Active Server Pages הם הפתרון המוצע על ידי Microsoft לבניית אתרי אינטרנט מתקדמים. רובם המכריע של אתרי האינטרנט הגדולים והמצליחים מבחינה עסקית הפועלים כיום ברשת האינטרנט - כגון אתר Dell (www.dell.com), Barnes and Noble (www.bn.com) ו-Microsoft עצמה (www.microsoft.com) - נבנו באמצעות דפי ASP. בספר זה תלמד כל שעליך לדעת כדי ליצור אתר אינטרנט במורכבות דומה.

השימוש בדפי ASP יאפשר לך:

- ליצור דפי אינטרנט דינמיים. דף ASP מאפשר לך להציג תוכן שונה למשתמשים שונים בזמנים שונים של היממה.
- עיבוד תוכן טפסי HTML. תוכל להשתמש בדף ASP כדי לאחזר ולהגיב על נתונים שהוקלדו לטופס HTML.
- יצירת דפי אינטרנט שעובדים בצורה אינטנסיבית עם מסדי נתונים. דף ASP יכול להכניס נתונים חדשים, או לאחזר נתונים קיימים, ממסד נתונים, כגון שרת SQL.
- לעקוב אחר שיח משתמש. תוכל להיעזר בדפי ASP כדי לאחסן מידע אודות משתמשים, מהרגע בו הגיעו לאתר שלך ועד לרגע בו הם עוזבים אותו.
- יצירת דפי אינטרנט המאפשרים חיפוש. תוך שימוש בחיפוש טקסט מלא של שרת אינדקס או שרת SQL, מאפשרים לך דפי ASP ליצור מנוע חיפוש לאתר האינטרנט שלך.
- לזהות את האפשרויות בדפדפנים השונים. דף ASP מסוגל לזהות את המאפיינים בהם תומך הדפדפן, ולהציג תוכן המתאים לדפדפנים השונים.
- שליחה וקבלת דואר אלקטרוני. דף ASP יכול לשלוח למשתמשים באתר הודעות דואר אלקטרוני באופן אוטומטי, ולאחזר דואר אלקטרוני הנשלח לאתר שלך.
- לשלב בין רכיבים מותאמים באתר האינטרנט שלך. תוכל להרחיב את תסריטי דפי ASP שלך באמצעות רכיבי צד-שרת מותאמים אישית, אותם תוכל ליצור באמצעות Microsoft Script Components, Visual Basic או ++Visual C.

# מדוע כדאי לך לרכוש ספר זה

מהדורה זו של הספר **המדריך השלם ASP 3** עודכנה בהתאם לשינויים האחרונים שנוספו לטכנולוגיית ASP על ידי חברת Microsoft.

- ספר זה מכיל שלושה יישומים לדוגמה מלאים של דפי ASP. תוכל ללמוד כיצד לבנות פורום דיונים, חנות מקוונת או אתר לחיפוש עבודה, תוך שימוש בתסריטי ASP (ASP Scripts) בלבד.
- שלושה פרקים מוקדשים ליצירת רכיבי ASP מותאמים אישית. בפרקים אלה תלמד כיצד לבנות רכיבים מותאמים אישית באמצעות Microsoft Script Components, Visual Basic או ++Visual C.
- ספר זה מסביר כיצד להשתמש בממשקי שירות הספריה הפעילה (ADSI, Active Directory Service Interfaces) מתוך דפי ASP שלך. תלמד כיצד להשתמש ב-ADSI ליצירת משתמשים וקבוצות בסביבת Windows NT, וכיצד לטפל בנתוני אבטחה.
- ארבעה פרקים בספר מוקדשים לגרסאות העדכניות ביותר של אובייקטי נתונים של ActiveX (ADO, ActiveX Data Objects) ו- OLE DB. למשל, תלמד כיצד ליצור ספקי OLE DB טבעיים וכיצד ליצור ערכות רשומות (Recordsets) מתמידות (Persistent) ושאינן מחוברות לבסיס נתונים (Disassociated).
- הספר מציג את המאפיינים החדשים שבגירסה 7.0 של שרת SQL. לדוגמה, תלמד כיצד להשתמש בחיפוש טקסט מלא של שרת SQL גירסה 7.0 מתוך דפי ASP שלך.
- תלמד אודות Microsoft DNA. תלמד כיצד שרת טרנזקציות (Transaction Server) ושרת תור ההודעות (Message Queue) גורמים לאתר האינטרנט שלך להיות גמיש יותר לצמיחה, ועמיד בפני תקלות.
- תסקור את המאפיינים החדשים בדפי ASP וב-IIS הנכללים כחלק ממערכת ההפעלה Windows 2000.

# למי מיועד הספר

ספר זה נכתב עבור המפתח המקצועי המייצר או מתחזק אתר אינטרנט או אינטראנט: Web Master, Web e-Commerce, Web Programmer, Web Builder. הספר מתמקד ביצירת אתרי אינטרנט דינמיים, ובעיקר באלה שבסיסם נמצא מסדי נתונים.

הספר יוצא מנקודת הנחה כי יש לקורא בו הבנה בנושא HTML ובנושא VBScript. בנוסף עליך להכיר היטב את מערכת ההפעלה Windows NT ואת שרת האינטרנט של חברת Microsoft IIS - Internet Information Server - Microsoft. לחיזוק נושאים אלה אתה מוזמן לעיין בספר **HTML 4 למפתחי אתרים באינטרנט** בהוצאת **הוד-עמי** הכולל גם פרק על VBScript.

לסיום, כדי להבין את הפרקים הדנים בנושא הגישה למסדי נתונים באמצעות דפי ASP רצוי שיהיה לך רקע כלשהו בשרתי SQL ובשפת SQL.

## English Please

- המונחים בספר מופיעים באנגלית ולידם הסבר בעברית.
- ASP Reference באנגלית,
- VBScript Reference באנגלית,
- SQL Reference באנגלית,
- האינדקס הענק, שבעזרתו תוכל למצוא כל דבר, כל אובייקט וכל שיטה, הוא באנגלית.

## הסימנים _ ⇘

בשפת VBScript יש לכתוב את משפטי הקוד ורק בסיום המשפט ללחוץ Enter. שורות ארוכות במיוחד ניתן "לשבור" על ידי הצבת קו תחתי _ בסיום השורה. זהו פרט עיצובי בלבד ואין לו שום קשר לביצוע הקוד. בספר זה אנו בחרנו לסמן את המשך השורה על ידי הסימן ⇘. סימן זה מורה שיש להמשיך את השורה הקודמת ברצף או "לשבור" אותה בעזרת _, כרצונך.

כאשר אתה רואה משפט שכזה:

```
myVar = "When in the Course of human events, " &
  ⇘ "it becomes necessary for one people to " &
  ⇘ "dissolve the political bands which have " &
  ⇘ "connected them with another, and to assume "
```

הוא יכול להיכתב כך, או בכל דרך אחרת שתרצה ובלבד שהמשפט "לא יישבר" על ידי Enter:

```
myVar = "When in the Course of human events, " &_
  "it becomes necessary for one people to " &_
  "dissolve the political bands which have " &_
  "connected them with another, and to assume "
```

## מונחים בעברית

במהלך הספר נעשה שימוש במילים **אחזור**, **לאחזר**, **מאוחזר** במשמעות של לקבל, לקרוא או להעביר (להביא). בחרנו להשתמש במילה זו מכיון ששום דבר חדש לא נוצר אלא נעשה שימוש במידע קיים. מקור המידע יכול להיות קובץ בשרת, ערך שדה במסד נתונים, ערך שדה בטופס או מאפיין באובייקט, זה כלל אינו משנה מהיכן הוא נלקח, החשוב הוא שהוא לא נוצר בתהליך. הוא כבר קיים ואנו עושים בו שימוש.

לדוגמה: כאשר אתה **מאחזר** קובץ משרת אינטרנט... .

את המשפט הזה ניתן לקרוא גם כך: כאשר אתה **קורא** קובץ משרת אינטרנט...

דוגמה נוספת: כשאתה מבקש דף HTML יכול הדפדפן שלך להחליט שלא **לאחזר** את הדף מאתר האינטרנט בו מתארח הדף. במקום זאת, הוא עשוי להחליט **לאחזר** את הדף המבוקש מעותק מוטמן (Cached Copy) השמור בדיסק הקשיח של המחשב שלך.

את המשפט הזה ניתן לקרוא גם כך: כשאתה מבקש דף HTML יכול הדפדפן שלך להחליט שלא **לקרוא** (במובן של העברת/הבאת מידע) את הדף מאתר האינטרנט בו מתארח הדף. במקום זאת, הוא עשוי להחליט **לקבל** (או לקרוא) את הדף המבוקש מעותק מוטמן (Cached Copy) השמור בדיסק הקשיח של המחשב שלך.

דוגמה: משתנה יישום (Application Variable) יכול לאחסן מידע **שאוחזר** מבסיס נתונים. למשל, אתה יכול **לאחזר** רשימת פריטים למכירה באתר שלך, ולהציג את הרשימה במספר דפים תוך שימוש במשתנה יישום (Application Variable).

את המשפט הזה ניתן לקרוא גם כך: משתנה יישום (Application Variable) יכול לאחסן מידע **שנקרא** מבסיס נתונים. למשל, אתה יכול **לקרוא** רשימת פריטים למכירה באתר שלך, ולהציג את הרשימה במספר דפים תוך שימוש במשתנה יישום (Application Variable).

המונחים **Function**, **Routine** תורגמו למילים **פונקציה** עבור Function ו- **שגרה** עבור Routine. באופן בסיסי הן פונקציה והן שגרה מבצעות את אותו הדבר, אם כי יש מספר שינויים. ההבדלים הם:

**שגרה** (Routine) יכולה לבצע פעולות, לחשב ערכים ועוד. יש שגרות המופעלות אוטומטית בעת התרחשות אירועים, אך תוכל להרחיב את השימוש בהן מעבר לכך. שגרות לעולם אינן מחזירות ערך או ערכים. שגרה מתחילה במילה Sub.

**פונקציה** (Function) אמורה לבצע משימה אחת, לכן ניתן לחלק את קוד התוכנית לקטעים קצרים. ניתן להעביר לפונקציה ערכים (parameters או arguments) בעת הקריאה לה. אז, תוכל להשתמש בערכים ולהציבם בתוך משתנים לצורך ביצוע חישוב ולבסוף להציב (או שלא להציב) את התוצאה שתוחזר בשם הפונקציה. פונקציה יכולה להחזיר ערך אך אינה חייבת לעשות כך. פונקציה מתחילה במילה Function. שפת VBScript מכילה הרבה מאוד פונקציות מובנות בתוכנה (Built-In functions). בנוסף, תוכל להגדיר פונקציות משלך (User-define functions).

ואז הגיע **Subroutine** ובעברית **שגרה**. גם המונח **Procedure** תורגם **לשגרה**. ההסבר לכל אותם מינוחים הוא שמקורם בשפות תכנות שונות. בשפה אחת קראו Routine ושפת תכנות אחרת אותו הדבר נקרא Subroutine ובשפה אחרת Procedure.

המונח **Validation** תורגם למילה **אימות** (מהמילה אמת) במשמעות של בדיקת תקינות הנתונים, שהנתונים הגיוניים. למשל, תהליך האימות של מחרוזת תווים כלשהי כדי לדעת אם היא מספר טלפון יכלול (בין השאר) חיפוש אחר תווים שאינם ספרות (אסור שבמספר טלפון ימצאו התווים & % ואחרים). בבדיקה זו לא נבדקת נכונות הנתונים. במקרה של מספר הטלפון: לא נבדק האם הטלפון הזה קיים ו/או המספר מחובר וכדומה. כוונת הבדיקה היא לוודא שהמספר הוא מספר טלפון חוקי (שאין לו,

לדוגמה, רק 5 ספרות) ושהתוכנית בהמשך תוכל להתייחס אליו כמספר טלפון. אין מניעה שתהליך האימות (validation) יכלול בדיקה מקיפה יותר שתבדוק מעבר לבדיקת תקינות בלבד. למשל, אם משתמש הקליד מספר טלפון סלולרי, תהליך האימות יכלול בדיקות למציאת תווים שאינן ספרות (דבר שאסור) וגם יבדוק שהקידומת היא בת 3 ספרות ושתי הספרות שמשמאל הן 05. כי אם הקידומת של הטלפון הסלולרי היא 09 זה לא חוקי.

המונח **Authentication** וגם המונח **Verification** תורגמו למונח **אימות** והפעם כדי לוודא שזהו **באמת** אתה ולא מישהו אחר. הפעם, המשמעות היא של **אישור**.

מסד נתונים ו- בסיס נתונים הם אותו דבר - database.

מסד נתונים מכיל רשומות. כל **רשומה** מכילה **שדות**. מכיון שאנו מדברים על מסדי נתונים טבלאיים, מתייחסים לאוסף של רשומות כאל **טבלה**, לרשומה בתור **שורה** ולשדה בתור **עמודה**.

הפועל **to Index** תורגם **למפתח** (מהמילה מפתח). לפעמים מונח זה תורגם לייצר אינדקס או לעדכן מחדש אינדקס.

המונח **Transaction** תורגם ל**תנועה**.

# כיצד בנוי הספר

הספר מאורגן כך שתוכל לקרוא בו מתחילתו ועד סופו. הפרקים הראשונים בספר דנים בנושאים בסיסיים בעיקרם. הפרקים אחריהם דנים בנושאים מתקדמים יותר. לפניך תקציר הנושאים הנידונים בכל חלק בספר:

חלק 1, אובייקטים של דפי ASP, דן באובייקטים המובנים הכלולים בדפי ASP. למשל, תלמד כיצד לעבוד עם נתונים המאוחזרים מטפסי HTML וממחרוזות שאילתה (Query Strings). בנוסף תלמד כיצד לאחסן נתונים מתמידים במשתנים Application ו-Session.

חלק 2, רכיבים ברי-התקנה של דפי ASP, דן במספר רכיבים נוספים הכלולים בדפי ASP. לדוגמה, תלמד כיצד להשתמש ב- CDO for NTS כדי לשלוח ולאחזר דואר אלקטרוני, ADSI כדי לטפל בנתוני אבטחה של משתמש וברכיב Ad Rotator כדי להציג כרזת פרסום (banner) שונה באתר האינטרנט שלך.

חלק 3, שימוש בדפי ASP עם מסדי נתונים, מלמד כיצד להשתמש ב-ADO לצורך גישה למסדי נתונים. בנוסף תלמד כיצד להשתמש ב-ADO עם שרת אינדקס (Index Server) וחיפוש טקסט מלא של שרת SQL גירסה 7.0.

חלק 4, רכיבים מותאמים של דפי ASP, דן כיצד יש ליצור רכיבי צד-שרת באמצעות Microsoft Script Components, Visual Basic או Visual C++. בנוסף תלמד בחלק זה אודות מספר רכיבי צד-שלישי יעילים אותם תוכל לכלול בדפי ASP שלך.

חלק 5, יצירת יישומים מבוזרים מדורגים, דן בנושא Microsoft DNA (Distributed interNet Applications Architecture), שרת הטרנזקציות של Microsoft, ושרת תור ההודעות.

חלק 6, נושאים מתקדמים, מתאר כיצד יש לתחזק את אתר האינטרנט שלך באמצעות Windows Scripting Host (WSH) ומתזמן המשימות (Task Scheduler). בנוסף תלמד בחלק זה כיצד למטב (optimize) את ביצועי אתר האינטרנט שלך, תוך שימוש ב-Profiler ו-InetMonitor. מעבר לכך, תלמד גם אודות המאפיינים החדשים של ASP הנכללים במערכת ההפעלה Windows 2000.

חלק 7, דוגמאות ליישומי ASP, דן כיצד יש ליצור שלושה אתרי אינטרנט מושלמים באמצעות דפי ASP. תלמד כיצד ליצור פורום דיונים, חנות מקוונת ואתר לחיפוש עבודה.

# התקליטור המצורף

כל קטעי הקוד המופיעים בספר זה, ניתנים בתקליטור המצורף. תוכל להשתמש בקטעים אלה בשלמותם כדי לחסוך זמן, ולמנוע טעויות הקלדה במהלך העבודה עם הדוגמאות שבספר.

התיקיה הרלוונטית לספר זה היא **X:\Books\59303** בה נמצאות תיקיות משנה לפי פרקי הספר. תוכל להעתיק את התיקיה לדיסק הקשיח שלך ולא לשכוח לבטל את הסימון **קריאה בלבד** ממאפייני התיקיה ו/או הקובץ.

**להסבר נוסף פנה בתחילת הספר (לאחר התוכן) ולקובץ ONCD שבתקליטור.**

# על היועץ המקצועי

**מוטי סופר**, מדריך בקורסי Web Master, Web Programming, Visual C++ ו-Java במרכזי הדרכה מובילים בישראל. משמש כמנהל פיתוח בחברת היי-טק.

# הערות, שאלות, רעיונות

לא נחסך כל מאמץ להבטחת הדיוק של ספר זה והתקליטור המצורף אליו. אם יש לך הערות, שאלות או רעיונות הנוגעים לספר זה של ההוצאה ולתקליטור המצורף, אנא שלח אותם להוצאת הוד-עמי באחת השיטות הבאות:

דואר אלקטרוני: info@hod-ami.co.il , בשורת Subject ציין את המספר 59303

דואר רגיל:

הוצאת הוד-עמי לספרי מחשבים

ת.ד. 6108

הרצליה 46160

# חלק 1

# אובייקטים של דפי ASP

**בחלק זה:**

- **בניית דפי ASP**
- **דפי ASP ופרוטוקול HTTP**
- **עבודה עם טפסי HTML**
- **עבודה עם יישומי דפי ASP**
- **עבודה עם Sessions**
- **פעולה עם הלקוח: ASP ו-Internet Explorer**

## פרק 1

# בניית דפי ASP

**בפרק זה:**

- **כיצד באמת פועלים דפי ASP**
- **שילוב אובייקטים ורכיבים בדפי ASP**
- **איתור וטיפול בתקלות בדפי ASP**

פרק זה יערוך לך היכרות רשמית עם דפי ASP (Active Server Pages). תלמד בו כיצד דפי ASP באמת פועלים, כיצד להשתמש באובייקטים וברכיבים בדפי ASP כאלה, וכיצד להגדיר, לאתר ולטפל בתקלות בעת התקנת דפי ASP בשרת אינטרנט (Web Server).

# כיצד באמת פועלים דפי ASP

דף ASP (Active Server Page) הוא קובץ HTML תקני ובו מספר מאפיינים נוספים. כמו בקובץ HTML תקני, דף ASP מכיל תגיות HTML היכולות להיות מפורשות ומוצגות על ידי דפדפן אינטרנט. כל מה שתוכל לשים בקובץ HTML תקני (למשל, יישומוני Java, טקסט מהבהב, תסריטי צד-לקוח, פקדי ActiveX צד-לקוח) תוכל לשים גם בדפי ASP. אולם, לדף ASP יש שלושה מאפיינים המייחדים אותו:

- דף ASP יכול לכלול תסריטי צד-שרת. על ידי הכללת תסריט צד-שרת בדף ASP אתה יכול ליצור דפי אינטרנט בעלי תוכן דינמי. כדי לתת דוגמה פשוטה ביותר, אתה יכול ליצור דף אינטרנט המציג הודעות שונות בהתאם לשעה ביום.
- דף ASP מספק מספר אובייקטים מובנים (Built-in Objects). על ידי השימוש באובייקטים מובנים הזמינים בדף ASP תוכל להעצים את כוחם של התסריטים אותם אתה יוצר. בין השאר, אובייקטים אלה מאפשרים לך הן לשלוח והן לקבל מידע אל ומאת דפדפנים. לדוגמה, תוך שימוש באובייקט Request אתה יכול לקבל מהמשתמש מידע המוצב בטופס בפורמט HTML ולהגיב למידע זה באמצעות תסריט.
- דף ASP יכול להיות מורחב באמצעות רכיבים אפשריים נוספים. דפי ASP מגיעים כשבהם נכללים מספר פקדי ActiveX מובנים של צד השרת. רכיבים אלה מאפשרים לך לבצע פעולות כגון עבודה עם מסדי נתונים, משלוח דואר אלקטרוני (E-Mail) וגישה למערכת הקבצים.
  רכיבי ActiveX תקניים אלה שימושיים במיוחד, אולם אינך מוגבל רק אליהם. באפשרותך לרכוש רכיבים של צד-שלישי או ליצור רכיבי ActiveX נוספים משלך. אין גבול ליכולותיך להרחיב דפי ASP.

שלושה מאפיינים אלה מגדירים את דפי ASP. דף ASP הוא דף HTML תקני אשר הורחב באמצעות תסריטי צד-שרת, אובייקטים ורכיבים. תוך שימוש בדפי ASP תוכל ליצור אתרי אינטרנט בעלי תוכן דינמי.

הדרך הטובה ביותר להבין כיצד פועלים דפי השרת היא על ידי הצגת ההבדלים שבין גישה לדף ASP לעומת הגישה לדף HTML רגיל.

תהליך שירות דף HTML רגיל יכול להיות מפורק לצעדים הבאים :

1. משתמש מקליד כתובת אינטרנט של קובץ HTML לשורת הכתובת (Address) בדפדפן ומקיש Enter כדי לבקש את דף האינטרנט (לדוגמה, http://www.aspsite.com/hello.htm).
2. הדפדפן שולח לשרת אינטרנט (Web Server), כגון IIS (Internet Information Server), בקשה לקבל את דף האינטרנט.
3. שרת האינטרנט מקבל את הבקשה ומזהה שהיא עבור קובץ HTML, מכיון שלקובץ המבוקש מצוינת סיומת HTM או HTML.
4. שרת האינטרנט מאתר את קובץ HTML הנדרש מהדיסק או מהזיכרון ושולח אותו חזרה לדפדפן.
5. קובץ HTML מפורש על ידי דפדפן המשתמש והתוצאות מוצגות בחלון הדפדפן.

כמובן שהתהליך מורכב מעט יותר, בדרך כלל (לדוגמה, תוכן הטפסים מוצג ומחרוזות שאילתה מועברות). אולם בגדול, צעדים אלה מתארים באופן כללי את הפעילות מרגע לרגע של שרת אינטרנט טיפוסי. שרת מקבל בקשות לקבצים מסוימים ומגיב על ידי שליחת הקובץ הנכון, כשהוא מוצא אותו בדיסק הקשיח או בזיכרון.

**הערה:**

הדיון בנושא דפי ASP בספר זה יוצא מנקודת הנחה שתשתמש בדפי ASP בסביבת שרת האינטרנט (IIS, Internet Information Server) של Microsoft. אולם, ניתן גם להשתמש בדפי ASP במיגוון רב של שרתי אינטרנט אחרים. דפי ASP יכולים לשמש גם בסביבת שרת האינטרנט האישי של Microsoft (PWS, Personal Web Server) עבור Windows 9x, ובסביבת Peer Web Server עבור Windows NT.

אינך מוגבל להשתמש גם בדפי ASP רק בשרתי האינטרנט של Microsoft. תוך שימוש בתוכנה בשם Chili!Soft מבית היוצר של Chili!ASP תוכל להשתמש בדפי ASP גם בשרתי Netscape Enterprise ו- FastTrack, Lotus Domino ושרת Go, O'Reilly's WebSite ושרתי אינטרנט רבים נוספים. למידע נוסף, בקר באתר האינטרנט של Chili!Soft בכתובת www.chilisoft.com.

למרות ש-IIS יכול לשרת דפי HTML סטטיים, דפי ASP מאפשרים ל-IIS לשרת גם תוכן דינמי. באמצעות דפי ASP אלה תוכל ליצור דפים בעלי תוכן חדש כתגובה לבקשות המשתמש. השרת עצמו הופך לפעיל בתהליך יצירת דף האינטרנט.

חשוב מאוד להבין את הניגוד שבתהליך שירות דף ASP לעומת שירות דף HTML רגיל, ולצורך כך נפרק אותו לשלבים :

1. המשתמש מקליד כתובת אינטרנט של דף ASP בשורת Address של דפדפן האינטרנט שלו, ומקיש Enter כדי להגיע לדף ASP (לדוגמה : http://www.aspsite.com/hello.asp).
2. הדפדפן שולח בקשה ל-IIS.

3. שרת האינטרנט מקבל את הבקשה ומזהה אותה כבקשה לקובץ דף ASP, מפני שלקובץ המבוקש יש את הסיומת asp.

4. השרת מאתר את הקובץ המבוקש בדיסק הקשיח או בזיכרון.

5. השרת שולח את הקובץ לתוכנית מיוחדת הנקראת ASP.dll.

6. קובץ דף ASP מעובד מראשיתו ועד סופו וכל פקודה שמזהה התוכנית המעבדת, מבוצעת. תוצאת התהליך היא קובץ HTML תקני.

7. קובץ HTML נשלח חזרה לדפדפן.

8. הדפדפן במחשבו של המשתמש מתרגם את קובץ HTML והתוצאות מוצגות בחלון הדפדפן.

**הערה:**

לשם ההבהרה, הצעדים המתוארים כאן הוצגו בפשטות יתירה. דפי ASP אינם צריכים להיות מהודרים מחדש (Recompiled) בכל פעם שהם מתבקשים. אם דף ASP כלשהו התבקש בעבר ולא נעשה בו שינוי, הוא ייטען מהמטמון (Cache) במקום שיעובד פעם נוספת.

מנקודת מבטו של שרת האינטרנט, דף ASP שונה מאוד מקובץ HTML רגיל. קובץ HTML רגיל נשלח לדפדפן מבלי שיהיה מעובד באופן כלשהו. לעומת זאת, כל הפקודות שבדף ASP חייבות להיות מופעלות מלכתחילה, כדי ליצור קובץ HTML. דבר זה הוא אשר מאפשר לדף ASP להכיל תוכן דינמי.

מנקודת מבטו של הדפדפן, לעומת זאת, דף ASP הוא כמעט וזהה לדף HTML רגיל. ההבדל היחידי הוא שדף ASP יהיה בעל סיומת asp, במקום סיומת htm או html. כאשר מתקבלת בקשה לדף ASP מקבל השרת דף HTML רגיל. דבר זה הוא אשר מאפשר לדף ASP להיות תואם לכל סוגי הדפדפנים.

# שילוב אובייקטים ורכיבים בדפי ASP

דף ASP הוא בעיקרו Scripting Environment. ספר זה יוצא מנקודת הנחה שתיצור את התסריטים עבור דפי ASP שלך באמצעות VBScript של Microsoft. אולם, באפשרותך להשתמש בדפי ASP גם בשפות תסריט אחרות, כגון Jscript או PerlScript. כל שפת תסריט כוללת Script Engine התואם לתקן תסריטי ActiveX היכול לשמש ליצירת דפי ASP.

**הערה:**

אם שפת Perl מוכרת לך, בוודאי תשמח לדעת כי קיים יישום שלה התואם לדפי ASP. למידע נוסף בקר באתר http://www.activestate.com, או בקר באתר המלווה ספר זה http://www.aspsite.com.

הדרך הקלה ביותר להוספת תסריט (Script) לדף ASP היא תוך שימוש בתוחמים (Delimiters) <% ו-%>. כל טקסט הנמצא בין שני תוחמים אלה יעובד כתסריט. להלן דוגמה:

```
<HTML>
<HEAD><TITLE>ASP Script</TITLE></HEAD>
<BODY>
This is a
<% FOR (i=1 TO 10 %>
very,
<% NEXT %>
very long sentence.
</BODY>
</HTML>
```

כאשר דף ASP זה מוצג בדפדפן אינטרנט יופיע המשפט הבא:

This is a very, very, very, very, very, very, very, very, very, very, very long sentence

תסריט זה יוצר 10 עותקים של המילה very תוך שימוש בלולאה FOR...NEXT של VBScript.

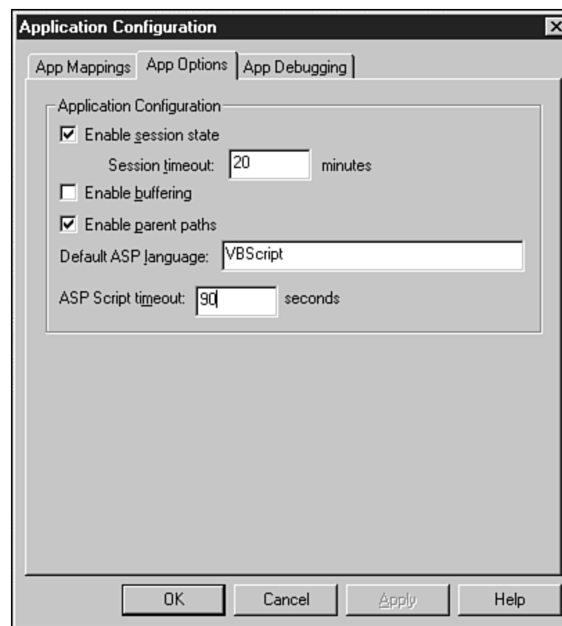
כברירת מחדל, דף ASP יוצא מנקודת הנחה שאתה משתמש ב-VBScript כשפת התסריט העיקרית שלך. דבר זה אומר שאינך צריך לעשות דבר מעבר לשימוש בתוחמי התסריט (Script Delimiters) <% ו-%> כדי להשתמש בשפת תסריט זו. אבל, קיימות שלוש דרכים לציון מדויק של השפה בה עושה דף ASP שימוש.

ראשית, תוכל להיעזר ב-Internet Service Manager כדי לציין שפת תסריט מסוימת כשפת תסריט שתהווה ברירת המחדל עבור דפי ASP שלך. כדי לעשות זאת, בצע את הצעדים הבאים:

1. פתח את תפריט Start, הצבע על Programs ומתפריט המשנה Microsoft Internet Information Server בחר את Internet Service Manager.
2. לחץ לחיצה ימנית על שרת האינטרנט שלך. אם לא שינית את תצורת ברירת המחדל יהיה שמו של השרת Default Web Site.
3. מתפריט הקיצור בחר Properties.
4. בחר בכרטיסיה Home.
5. לחץ על לחצן Configuration (כדי לעשות זאת צריך להיות לך יישום קיים. אם אין לך כזה, צור אחד חדש על ידי לחיצה על לחצן Create).

6. בתיבת הדו-שיח Application Configuration, בחר בכרטיסיה App Options.

7. בתיבת הטקסט Default ASP language הקלד את שם שפת התסריט בה אתה מעוניין להשתמש כשפת ברירת המחדל שלך ; לדוגמה, הקלד VBScript או JScript (ראה תרשים 1.1).




**תרשים 1.1** השימוש ב-Internet Service Manager לציון שפת תסריט ברירת מחדל

לאחר שהגדרת שפת תסריט מסוימת כברירת המחדל לתסריטים שלך, תוכל להשתמש בה בדפי ASP שלך, פשוט על ידי השימוש בתוחמים %> ו-<%. אם, למשל, אתה מתעתד להשתמש בעיקר בשפת JScript לדפי השרת שלך, עליך להגדיר שפה זו כשפת ברירת המחדל.

תוכל לציין גם את שפת התסריט העיקרית עבור דף מסוים. כדי לעשות זאת, הצב את המנחה LANGUAGE בשורה הראשונה בקובץ הדף השרת שלך, כפי הנראה בדוגמה הבאה :

```
<%@ LANGUAGE=JScript %>
<HTML>
<HEAD><TITLE>ASP Script</TITLE></HEAD>
<BODY>
This is a
<% for (i=1;i<11;i++) { %>
very,
<% } %>
very long sentence.
</BODY>
</HTML>
```

המנחה בשורה הראשונה של תסריט זה מציין שכל התסריטים הנכללים בקובץ זה צריכים להיות מופעלים כתסריטים שנוצרו באמצעות JScript.

```
<HTML>
<HEAD><TITLE>ASP Script</TITLE></HEAD>
<BODY>
<SCRIPT LANGUAGE="JScript"  RUNAT="server">
function sayhello()
{
response.write("Hello!")
}
</SCRIPT>
This is a
<%
sayhello()
%>
</BODY>
</HTML>
```

בדוגמה זו, התגית <SCRIPT> מכילה פונקציית JScript. המאפיין LANGUAGE של התגית <SCRIPT> מציינת באיזו שפת תסריט יש להשתמש. המאפיין RUNAT מציין שהתסריט צריך להיות מבוצע בשרת, ולא בלקוח (הדפדפן).

הפונקציה ()sayhello מגדירה את התסריט הראשון. התסריט השני, מסומן באמצעות התוחמים (Delimiters) %> ו-<% הרגילים, הוא המקום בו בעצם נקראת פונקציית JScript. דף ASP זה מציג את הכתובת !Hello על המסך.

אם יצרת תסריט JScript או VBScript של צד-לקוח, עליך להכיר את התגית <SCRIPT>. Microsoft הרחיבה תגית זו כך שתוכל לשמש לציון תסריט צד-שרת או צד-לקוח. אם לא הוספת את המאפיין "RUNAT="server בדוגמה הקודמת, תהיה ההתייחסות לתסריט כאל תסריט צד-לקוח. במקרה כזה, השרת יתעלם מהתסריט והדפדפן ינסה להפעיל אותו (הוא ייכשל כישלון מוחץ, מכיון שהתסריט אינו תסריט צד-לקוח תקני).

מדוע תרצה להשתמש אי פעם בתגית <SCRIPT>, ולא בתוחמים %> ו-<% הרגילים? בדרך כלל, לא תעשה כן. אולם, קיימים שני הבדלים מהותיים בין שתי הדרכים שתוארו לציון תסריט.

ראשית, תסריטים הנכללים בתגית <SCRIPT> מופעלים מייד, ואין זה משנה היכן הם מופיעים בדף ASP. לדוגמה, ראה את הדף הבא:

```
<HTML>
<HEAD><TITLE>ASP Script</TITLE></HEAD>
<BODY>
This is the first sentence.
<SCRIPT LANGUAGE="JScript"  RUNAT="server">
response.write("This is the second sentence.")
</SCRIPT>
</BODY>
</HTML>
```

במבט על התסריט, תוכל להתפתות ולהאמין שהמשפט This is the first sentence והמשפט "This is the second sentence." יוצגו על המסך בסדר הזה. אולם, כאשר דף ASP מוצג בדפדפן, מתהפך סדר הצגת המשפטים (או אף גרוע מכך, לא מוצג על המסך דבר, מפני שנוצר קובץ HTML לא חוקי).

מדוע זה קורה? כל מה שנמצא בין התגיות <SCRIPT> מופעל לפני כל דבר אחר בדף. אם תשתמש בפקודה View Source של הדפדפן שלך תראה את הדף הבא, שהוא תוצאה של תהליך הידור דף ASP הקודם:

```
This is the second sentence.<HTML>
<HEAD><TITLE>ASP Script</TITLE></HEAD>
<BODY>
This is the first sentence.
</BODY>
</HTML>
```

להתנהגות זו של תגית <SCRIPT> יש שתי השלכות. ראשית, אתה יכול להציב את התסריטים הנכללים בין תגיות <SCRIPT> בכל מקום בדף ASP שלך. שנית, התגית <SCRIPT> מוגבלת (ברוב המקרים) להכלת פונקציות ופרוצדורות בלבד. הפלט של כל תסריט שאינו מוכל בפונקציה או פרוצדורה מוצג מייד, והתוצאה היא דף HTML לא חוקי.

לתגית <SCRIPT> יש יתרון בולט אחד על תוחמי התסריט <% ו-%>. תוך שימוש בתגית <SCRIPT> אתה יכול לערבב מספר שפות תסריט בדף ASP אחד. ראה את הדוגמה הבאה:

```
<%@ LANGUAGE=JScript %>
<HTML>
<HEAD><TITLE>ASP Script</TITLE></HEAD>
<BODY>
<SCRIPT LANGUAGE="JScript"  RUNAT="server">
function sayhello()
{
response.write("Hello!")
}
</SCRIPT>
<%
FOR (i=1 TO 10
sayhello()
NEXT
%>
</BODY>
</HTML>
```

תסריט זה מציג 10 פעמים את המילה Hello!, בשורה. אך שים לב כיצד הדבר מתבצע. התסריט שבין התוחמים <% ו-%> הוא תסריט של Visual Basic Script. אולם, התסריט קורא לפונקציית JScript. פונקציית JScript מוגדרת בין תגיות <SCRIPT>. כשאתה מעוניין להשתמש בשפת תסריט כלשהי כשפה העיקרית שלך, אך עליך לקרוא

לפונקציה משפת תסריט אחרת, תוכל לעשות זאת בשיטה זו. אפשרות זו יעילה כאשר לשפה מסוימת יש פונקציה או שיטה (Method) מסוימת שבשפת ברירת המחדל שלך חסרה.

לסיכום, קיימות שלוש שיטות לכלול תסריט בדף ASP:

- ציין שפת תסריט לכל דפי ASP שאתה יוצר תוך שימוש ב-Internet Service Manger לציון שפת תסריט שתהווה את ברירת המחדל.
- ציין שפת תסריט עבור דף בודד, תוך שימוש במנחה (Directive) דפי ASP <%@ LANGUAGE="scripting language" %>.
- ערב מינים שונים של שפות תסריט בדף ASP בודד, על ידי שימוש בתגית <SCRIPT> המורחבת.

לפני שנסיים חלק זה, עלינו לדון במנחה דפי ASP נוסף.

תוך שימוש במנחה הפלט (Output Directive) של דפי ASP ניתן להציג את ערכו של ביטוי. הנה דוגמה:

```
<HTML>
<HEAD><TITLE>ASP Example</TITLE></HEAD>
<BODY>
At the tone, the time will be: <%=TIME%>
</BODY>
</HTML>
```

אתה משתמש במנחים <%= ו- %> כדי להדפיס ערך של משתנה, שיטה או פונקציה. בדוגמה הקודמת, מנחה הפלט משמש להצגת ערכה של פונקציית VBScript בשם TIME.

קיימת דרך נוספת להשגת אותה תוצאה. ראה את הדוגמה הבאה:

```
<HTML>
<HEAD><TITLE>ASP Example</TITLE></HEAD>
<BODY>
At the tone, the time will be: <% Response.Write(TIME) %>
</BODY>
</HTML>
```

בדוגמה זו, ערך הפונקציה TIME של VBScript מוצג תוך שימוש באובייקט Response של דף ASP. השיטה ()Write של האובייקט Response מציגה את ערך הביטויים על המסך (בחלק הבא תלמד עוד אודות השימוש באובייקטים).

מתי עליך להשתמש בשיטה Response.Write, ולאו דווקא במנחים <%= ו-%>? זה לא ממש משנה. בכל מקרה מייצגים דפי ASP את מנחי הפלט, באופן פנימי, כאילו היו קריאה לשיטה Response.Write. ניתן להשתמש בכל אחת משתי השיטות להצגת ערכי ביטויים.

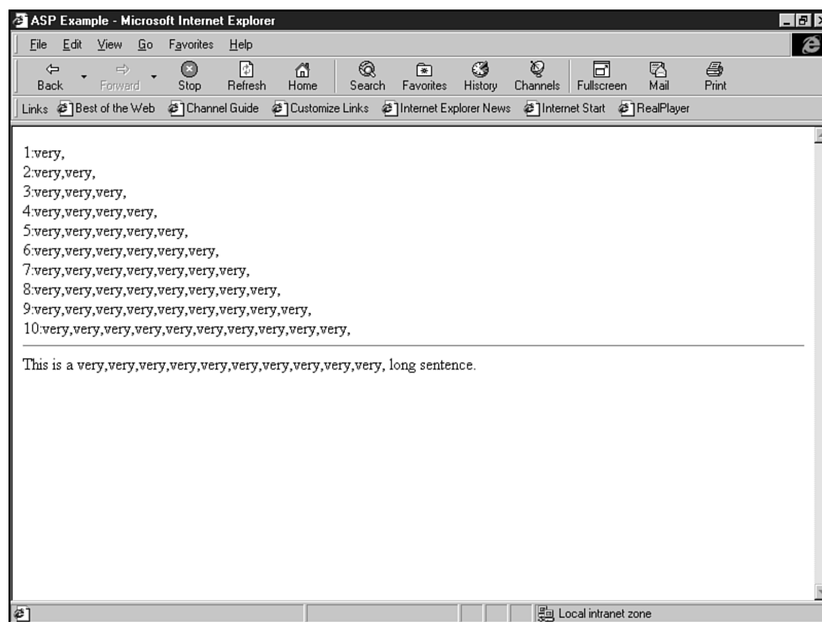
**הערה:**

כאשר אתה מקבל הודעת שגיאה בעת השימוש במנחי הפלט <%= ו-%>, תתייחס הודעת השגיאה לשורת קוד המכילה את השיטה ()Response.Write. הדבר יכול להיות מבלבל, מפני ששורה זו של הקוד בעצם אינה קיימת בדף ASP שיצרת. אולם, מצב כגון זה מציין בפניך ש-ASP משתמש לחילופין בשתי השיטות להצגת אותם ערכי ביטויים.

מעבר לכך, קיימים מצבים בהם שיטה אחת של הצגת ערכי ביטויים נוחה יותר מהאחרת. לדוגמה, כשאתה זקוק להצגת ערך ביטוי בתוך בתסריט, השיטה Response.Write() קלה יותר לשימוש, בדרך כלל. מצד שני, כשאתה מעוניין להציג את ערך הביטוי בחלק של קוד HTML, יהיו המנחים (Directives) =%> ו-<% קלים יותר. הדוגמה הבאה מתארת את שתי הגישות:

```
<HTML>
<HEAD><TITLE>ASP Example</TITLE></HEAD>
<BODY>
<%
FOR i=1 TO 10
myvar=myvar&"very,"
Response.Write(i&" : " &myvar&"<BR>")
NEXT
%>
<HR>
This is a <%=myvar%> long sentence.
</BODY>
</HTML>
```

בדוגמה זו, השיטה Response.Write() משמשת בלולאה להצגת ערך המשתנה ששמו myvar כאשר גודלו גדל והולך. מנחי הפלט =%> ו-<% מוטבעים בקוד HTML רגיל. מנחה הפלט משמש להצגת ערך המשתנה myvar (ראה תרשים 1.2).



**תרשים 1.2** שתי שיטות להצגת ערכים

# הבנת אובייקטים ורכיבים

דפי ASP כוללים מספר אובייקטים מובנים ורכיבי ActiveX ברי התקנה. אובייקטים ורכיבים אלה יכולים לשמש להרחבת כוחם של תסריטי דפי ASP. מהם אובייקטים (Objects) ורכיבים (Components)?

אובייקט (Object) הוא דבר שבדרך כלל מכיל שיטות (Methods), מאפיינים (Properties) או אוספים (Collections). שיטה של אובייקט קובעת מה ניתן לעשות בעזרת אובייקט זה. מאפייני אובייקט יכולים להיקרא או להגדיר את מצבו של האובייקט. אוספים של אובייקט מייצרים קבוצות שונות של צמדי מפתחות וערכים (Key & Value pairs) המשויכים לאובייקט.

ניעזר בדומה מחיי היום יום. הספר טום סוייר (Tom Sawyer) הוא דוגמה לאובייקט. האובייקט מכיל שיטות (Methods) שונות הקובעות מה ניתן לעשות באמצעותו. לדוגמה, אתה יכול לקרוא את הספר, להשתמש בו כמעצור לדלת (שלא תיטרק ותינעל) ואפילו, אם אתה מרגיש אכזרי במיוחד, לקרוע אותו לגזרים. לאובייקט יש מספר מאפיינים (Properties). לדוגמה, יש בו מספר מסוים של עמודים והוא נכתב על ידי סופר מסוים. לסיום, יש לו אוסף (collection) של צמדי מפתחות וערכים. לכל מספר עמוד (מפתח) יש דף טקסט תואם (Value).

**רכיב** ActiveX (Component) דומה לאובייקט מובנה של דף ASP. אולם, כאשר אתה משתמש בדפי ASP, קיים הבדל משמעותי אחד בין רכיב לאובייקט. **מופע** (Instance) של רכיב חייב להיווצר קודם לאפשרות השימוש בו.

# אובייקטים של דפי ASP

דפי ASP כוללים מספר אובייקטים מובנים (Built-in Objects). אובייקטים אלה מאפשרים לך להגדיל את עוצמתם של התסריטים אותם אתה יוצר. לדוגמה, תוך שימוש באובייקטים אלה אתה יכול להשיג גישה לבקשות הדפדפן ולשלוט בדרך בה השרת מגיב לבקשות אלו. האובייקטים המובנים גם יאפשרו לך שליטה ב-User Session וביישומי שרת אינטרנט (Web-server Applications).

כבר היכרת דוגמה אחת לאובייקט מובנה, האובייקט Response. תוכל להשתמש באובייקט זה כדי לשלוח פלט לדפדפן. אבל, לאובייקט Response יש מספר מאפיינים, שיטות ואוספים חשובים נוספים.

בהמשך תמצא פירוט כיצד להשתמש בכל אחד מהאובייקטים המובנים. הרשימה הבאה מספקת סקירה כללית מהירה של כל אחד מהם:

- **Application** - האובייקט Application משמש לאחסון ולאחזור מידע שיכול להיות משותף בין כל משתמשי היישום. לדוגמה, אתה יכול להשתמש באובייקט זה כדי להעביר מידע בין משתמשים באתר האינטרנט שלך.

- **Request** - האובייקט Request יכול לשמש לשם גישה לכל המידע הנשלח בבקשה מדפדפן לשרת שלך. אתה יכול להשתמש באובייקט זה כדי לאחזר את המידע שהמשתמש הקליד בטופס HTML.
- **Response** - האובייקט Response משמש לשליחת מידע חזרה לדפדפן. אתה יכול להיעזר באובייקט זה כדי לשלוח ערכים מהתסריט שלך לתצוגה בדפדפן.
- **Server** - האובייקט Server מאפשר לך להשתמש בפונקציות שירות שונות בשרת. לדוגמה, אתה יכול להשתמש באובייקט כדי לשלוט במשך הזמן בו יופעל תסריט לפני שיפוג זמנו. אתה יכול גם להשתמש באובייקט זה כדי ליצור מופעים של אובייקטים אחרים.
- **Session** - האובייקט Session יכול לשמש לאחסון ולאחזור מידע אודות Session משתמש מסוים. אתה יכול להשתמש באובייקט כדי לאחסן מידע הנמשך בהתמדה לאורך ביקורו של המשתמש באתר האינטרנט שלך.
- **ObjectContext** - האובייקט ObjectContext משמש לשליטה ב-Transactions (תנועות) דפי ASP. התנועות מנוהלות על ידי שרת תנועות של Microsoft (MTS, Microsoft Transaction Server).

האובייקטים המובנים שונים מאובייקטים רגילים. אינך צריך ליצור מופע לאובייקט מובנה לפני שתוכל להשתמש בו בתסריט (script). השיטות, אוספים ומאפיינים של אובייקט מובנה נגישים באופן אוטומטי מכל יישום שרת אינטרנט.

**הערה:**

ראה בנספח B אובייקטים ורכיבי ASP המתייחסים לרשימה שלמה של השיטות, המאפיינים והאוספים של האובייקטים המובנים.

# רכיבי דפי ASP

כמו האובייקטים המובנים בהם דן החלק הקודם, רכיבי דפי ASP יכולים לשמש להעצמת כוחם של התסריטים אותם אתה יוצר. ההבדל בין רכיבים לאובייקטים מובנים נובע מכך שהם בדרך כלל משמשים למשימות ייעודיות יותר. הרשימה הבאה מספקת סקירה כללית מהירה של הרכיבים המאוגדים עם דפי ASP:

- הרכיב **Ad Rotator** - רכיב זה משמש להצגת פרסומי כרזה (Banner Advertisement) בדפי אינטרנט המוצגים באתר אינטרנט. אתה יכול להשתמש ברכיב זה לציון משך הזמן בו תוצג כל כרזת פרסום.
- הרכיב **Browser Capabilities** - רכיב זה יכול לשמש אותך להצגת תוכן HTML שונה, בהתאם ליכולותיהם של הדפדפנים השונים. לדוגמה, תוכל להשתמש ברכיב כדי להציג דפי אינטרנט במסגרות, אבל רק בדפדפנים התומכים בכך.

- הרכיב **Content Linking** - תוך שימוש ברכיב זה ניתן לקשר בין מספר קבצי HTML, כך שהניווט ביניהם יהיה קל. לדוגמה, השתמש ברכיב זה כדי להציג דפיו של ספר מקוון.
- הרכיב **Counters** - רכיב זה מאפשר לך לעקוב אחר מספר המבקרים באתר שלך. אתה יכול להוסיף רכיב זה כדי ליצור מונה כניסות (Hit Counter) לדף אינטרנט מסוים.
- הרכיב **Content Rotator** - רכיב זה מאפשר לך ליצור תוכן HTML משתנה בדף אינטרנט. לדוגמה, אתה יכול להיעזר ברכיב זה כדי להציג באופן אקראי הודעות שונות בדף הבית של אתר האינטרנט שלך.
- הרכיב **Page Counter** - זהו רכיב זהה לרכיב Counter שתואר קודם לכן, בכך שהוא יכול לשמש למעקב אחר מספר המבקרים ולהוסיף מונה כניסות לדף אינטרנט מסוים.
- הרכיב **Permission Checker** - רכיב זה יכול לשמש להצגת קישורים לדפי אינטרנט מסוימים רק למשתמשים להם יש הרשאה לצפות בהם. תוכל להיעזר ברכיב זה כדי ליצור דפי אינטרנט הניתנים לצפייה רק על ידי מנהלי האתר (Web Site Administrators).
- האובייקטים **Collaboration Data** - נקראים בקיצור CDO. אלה מאפשרים לך לשלוח ולקבל דואר אלקטרוני (E-Mail) מתוך דפי ASP שלך. אתה יכול להיעזר באובייקטים אלה כדי לשלוח הודעת דואר אלקטרוני למשתמש שזה עתה ביצע הרשמה לאתר שלך.
- האובייקטים **ActiveX Data** - נקראים בקיצור ADO. אלה מאפשרים לך לאחזר ולאחסן מידע במסד נתונים כדוגמת Microsoft SQL Server. אובייקטים אלה נחשבים לחשובים במיוחד. מסיבה זו הם נידונים בהרחבה בחלק נפרד של ספר זה, חלק 3, השימוש בדפי ASP עם מסדי נתונים.

חלק 2 של הספר, רכיבים ברי התקנה של ASP, מתאר בפירוט כיצד ניתן להשתמש בכל אחד מהרכיבים ברי ההתקנה. חלק 4 של הספר, רכיבי ASP מותאמים אישית, מתאר כיצד תוכל להרחיב את דפי ASP באמצעות רכיבים חדשים אותם תרכוש מחברות צד-שלישי או שתיצור בעצמך.

**הערה:**

ראה בנספח B אובייקטים ורכיבי ASP המתייחסים לרשימה שלמה של השיטות, המאפיינים והאוספים של הרכיבים.

# השימוש ברכיבים בדפי ASP

לפני שתוכל להשתמש ברכיב עליך, קודם כל, ליצור עבורו **מופע** (Instance). למרות שאתה יכול להיכנס באופן אוטומטי למאפיינים, לשיטות ולאוספים של האובייקטים המובנים בכל דף, לפני שתוכל להשתמש ברכיב עליך ליצור עבורו מופע עם טווח (Scope) ייחודי. בשלושת הסעיפים הבאים תלמד כיצד ליצור מופע של רכיב בטווח דף (Page), Session ויישום (Application).

## יצירת רכיב עם Page Scope

ברוב המקרים אתה תיצור מופע של רכיב עם טווח דף (Page Scope). רכיב עם טווח דף נוצר בדף יחיד וגווע כאשר עיבוד הדף מסתיים. אתה יכול להשתמש ברכיב עם טווח דף רק בדף בו הוא נוצר במפורש. כדי ליצור מופע עבור רכיב עם טווח דף אתה יכול להשתמש בשיטה ()Server.CreateObject או בתגית HTML בשם <OBJECT>.

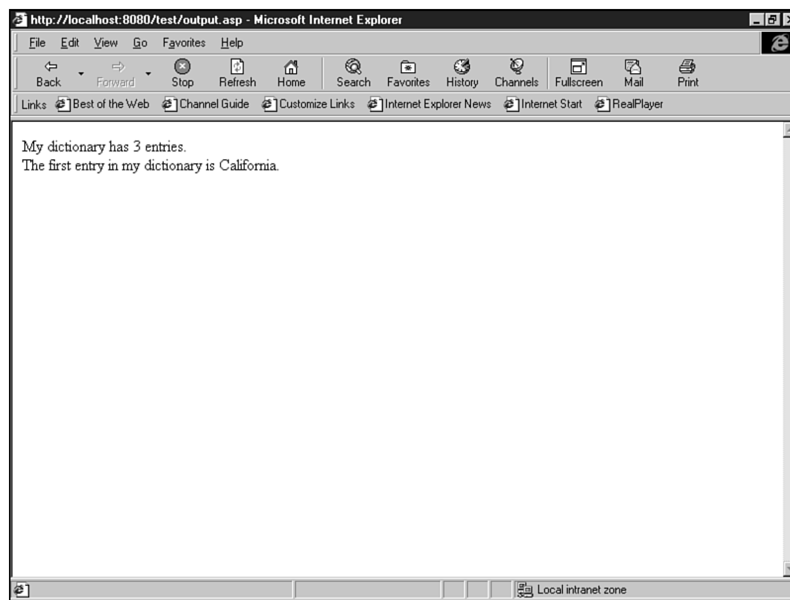
**אזהרה:**

אל תשתמש בשיטה CreateObject של VBScript או בהצהרה new של JScript כדי ליצור מופע של רכיב. אם תשתמש באחת משתי השיטות הללו ליצירת רכיב עלולים להתרחש דברים משונים ומאוד בלתי צפויים. במקום זאת, השתמש תמיד בשיטה ()Server.CreateObject של דפי ASP.

להלן דוגמה לאופן בו עליך להשתמש בשיטה CreateObject ליצירת מופע של האובייקט Dictionary עם טווח דף:

```
<%
Set MyDict=Server.CreateObject("Scripting.Dictionary")
MyDict.Add "CA", "California"
MyDict.Add "MA", "Massachusetts"
MyDict.Add "MI", "Missouri"
%>
My dictionary has <%=MyDict.Count %> entries.
<BR>
The first entry in my dictionary is <%=MyDict.Item("CA")%>.
```

כאשר תסריט זה מופעל נוצר מופע של האובייקט Dictionary. אחר כך, נוספים לו שלושה צמדי מפתח/ערך. לסיום, מוצגים שני מאפיינים של המילון (ראה תרשים 1.3).




**תרשים 1.3** דוגמה של האובייקט Dictionary

ההצהרה הראשונה בדוגמה זו משתמשת בשיטה Server.CreateObject כדי ליצור מופע של אובייקט עם טווח דף. בדוגמה זו משויך אובייקט המופע Dictionary למשתנה בשם MyDict.

לאחר שנוצר מופע האובייקט ניתן לקרוא לשיטות שלו. בדוגמה זו, נקראת השיטה Add של האובייקט Dictionary כדי להוסיף רשומות למילון. הקריאה הראשונה לשיטה Add משמשת להוספת הצמד מפתח/ערך (Key/Value) של CA עם California.

לאחר שנוצר מופע האובייקט אתה יכול גם לגשת למאפיינים שלו. בדוגמה זו המאפיין Count של האובייקט Dictionary נקרא כדי לקבוע את מספר הרשומות במילון. המאפיין Item נקרא גם הוא, כדי להחזיר את ערכו של מפתח מסוים.

בסופו של דבר, כשסיימת להשתמש במופע של אובייקט, אתה יכול להרוס (Destroy) אותו. אובייקט שנוצר על ידי שיטה קודמת ייהרס באופן אוטומטי אחרי שהשרת מסיים את עיבוד דף ASP. בשל כך, בדרך כלל לא תהיה סיבה להרוס אובייקט באופן מכוון. אבל, אתה יכול להרוס את האובייקט MyDict במכוון על ידי שיוך MyDict לערך חדש, או על ידי קביעת ערך המשתנה ל-Nothing, כך:

```
<% Set MyDict=Nothing %>
```

דוגמה זו מתארת את השיטה הנפוצה ביותר ליצירה ושימוש ברכיב ActiveX בדפי ASP. אתה יוצר מופע עבור אובייקט על ידי קריאה לשיטה ()Server.CreateObject. לאחר שאתה יוצר אובייקט חדש אתה יכול לקרוא לשיטות שלו ולקרוא או להגדיר מאפיינים.

**הערה:**

ראה בנספח B דוגמאות כיצד ליצור רכיבי ActiveX.

קיימת גם שיטה נוספת ליצירת אובייקטים עם טווח דף (Page Scope). במקום להשתמש בשיטה CreateObject אתה יכול ליצור מופע עבור האובייקט באמצעות התגית <OBJECT> המורחבת של Microsoft. הנה דוגמה לכך:

```
<OBJECT RUNAT="server" ID="MyBrow"
PROGID="MSWC.BrowserType"></OBJECT>
```

מאחר והקוד המופיע כאן אינו תסריט, אל לך להוסיף אותו לדף ASP בין תוחמי התסריט %> ו-<%. התגית <OBJECT> היא תגית HTML ועליה להופיע בחלק קוד HTML רגיל בדף שלך.

דוגמה זו מציגה כיצד להשתמש בשלושה מאפיינים של התגית <OBJECT>. המאפיין ID מספק לרכיב מזהה ייחודי (Unique Identifier), שם, כדי שתוכל להתייחס אליו בתסריטים בתוך דפי ASP שלך. המאפיין PROGID משמש לציון שמו הרשום של הרכיב. זה השם בו משתמש השרת לזיהוי הרכיב, כאשר הוא יוצר את המופע עבורו. זהו גם אותו שם שעליך להעביר לשיטה ()Server.CreateObject.

מתי עליך להשתמש דווקא בתגית <OBJECT>, ולאו דווקא בשיטה ()Server.CreateObject, כדי ליצור מופע עבור אובייקט? היתרון האחד של התגית <OBJECT> הוא בכך שהיא אינה יוצרת מופע עבור האובייקט עד אשר מאפיין או שיטה של האובייקט נקראים בפועל. אם דף ASP מכיל אובייקטים אשר ייתכן ולעולם לא ייקראו, עליך להשתמש בתגית <OBJECT> כדי לחסוך בזיכרון.

Microsoft ממליצה שתיצור את מרבית הרכיבים שלך באמצעות טווח דף. יצירת רכיבים באמצעות טווח דף (Page Scope) גורמת לכך שאתה מטיל מעמסה קלה יותר על שרת האינטרנט. רכיב טווח דף משחרר כל זיכרון ומשאבים אחרים להם הוא נדרש כאשר מסתיים תהליך עיבוד הדף.

## יצירת רכיב עם Session Scope

במקום ליצור רכיבים הנגישים רק בדף ASP בודד, אתה יכול ליצור רכיבים הנגישים במספר דפי ASP. דרך אחת לעשות כן היא יצירת רכיב עם Session Scope. רכיב עם Session Scope זמין עבור מבקר יחיד באתר האינטרנט שלך בכל דף שהמבקר מבקש. מופע של הרכיב נוצר עבור כל מבקר. מופע הרכיב אינו עוזב את הזיכרון עד שהמבקר עוזב את האתר.

ליצירת דף עם Session Scope קיימות שתי שיטות. שיטה אחת משייכת רכיב למשתנה Session (Session Variable) על ידי שימוש בשיטה ()Server.CreateObject, כפי שניתן לראות בדוגמה הבאה:

```
<%
Set Session ("MyBrow")=Server.CreateObject(MSWC.BrowserType)
%>
```

תסריט זה משייך מופע של הרכיב Browser Capabilities למשתנה Session בשם MyBrow. אתה יכול להציב תסריט זה בכל מקום בדף ASP או בקובץ Global.asa (תלמד אודות קובץ זה בפרק 4). בדרך כלל תמצא שהגיוני ליצור את מופע האובייקט בשגרה

Session_OnStart של הקובץ Global.asa, מפני שעליך ליצור את המופע פעם אחת בלבד עבור כל מבקר באתר.

### הערה:

המילה **שגרה** היא תרגום המונח **Subroutine**.

לאחר שיצרת רכיב עם Session Scope אתה יכול להשתמש במאפיינים ובשיטות שלו על ידי התייחסות למשתנה ה-Session המכיל את הרכיב. לדוגמה, התסריט הבא מציג את הערך של המאפיין Frames של הרכיב Browser Capabilities שנוצר בתסריט הקודם:

```
<%
if session("MyBrow"),Frames = TRUE then
        response.write "Your Browser Supports Frames!"
end if
%>
```

לשיטה זו של יצירת רכיב עם Session Scope יש חיסרון משמעותי. השיטה Server.CreateObject() יוצרת מופע עבור הרכיב באופן מיידי. אפילו אם לעולם לא נעשה שימוש במופע הרכיב, הוא עדיין גוזל משאבים מהשרת. מכיון שמופע חדש של אובייקט Session נוצר עבור כל מבקר, יכולה כמות המשאבים הנגזלים מהשרת להיות משמעותית מאוד.

למרבה המזל, קיימת דרך נוספת ליצירת רכיב עם Session Scope. ניתן להשתמש בתגית <OBJECT> כדי ליצור רכיב, הנה כך:

```
<OBJECT RUNAT="Server" SCOPE="Session" ID="MyBrow"
PROGID="MSWC.BrowserType"></OBJECT>
```

כשאתה יוצר מופע עבור רכיב עם Session Scope באמצעות התגית <OBJECT> עליך ליצור את המופע בתוך הקובץ Global.asa. צור את מופע האובייקט מחוץ לתת-השגרות Application_OnStart, Application_OnEnd, Session_OnStart ו- Session_OnEnd.

בדוגמה זו משתמשים בארבעה מאפיינים של התגית <OBJECT>. המאפיין SCOPE מציין שלרכיב שנוצר צריך להיות Session Scope. המאפיין ID מספק לרכיב את המזהה הייחודי שלו (שם), כדי שתוכל להתייחס אליו מתסריטי הנמצאים בדפי ASP. המאפיין PROGID משמש לציון שמו הרשום של הרכיב. זהו השם בו משתמש השרת כדי לזהות את הרכיב כאשר הוא יוצר את המופע עבורו. זה אותו שם שעליך להעביר לשיטה Server.CreateObject().

### הערה:

כשאתה משתמש בתגית <OBJECT>, במקום להשתמש בשמות הרשומים (PROGID) של הרכיבים אתה יכול להשתמש במספרים הרשומים (CLASSID). הרי דוגמה:

```
<OBJECT RUNAT="Server" SCOPE="Session" ID="MyBrow"
CLASSID="0ACE4881 - 8305 - 11CF - 9427 - 444553540000"></OBJECT>
```

כשאתה יוצר מופע עבור רכיב מתוך התגית <OBJECT> אתה יכול לגשת לכל אחת מהשיטות ומהמאפיינים שלו כאילו הוא היה רכיב מובנה של דפי ASP. לדוגמה, התסריט הבא מציג את מאפייני הרכיב Browser Capabilities (הוא יוצא מנקודת הנחה שמופע הרכיב נוצר בקובץ Global.asa).

```
<%
if MyBrow.Frames = TRUE then
        response.write "Your Browser Supports Frames!"
end if
%>
```

כאשר הרכיב נוצר עם Session Scope באחת משתי הדרכים שתוארו להלן, כל אחד מהאוספים, השיטות או המאפיינים זמינים לכל דף אותו מבקש משתמש מסוים. אולם, יש ליצור מופע ייחודי של הרכיב עבור כל משתמש. רכיב עם Session Scope נוצר באופן יחסי ל-Session של משתמש מסוים.

מתי תצטרך ליצור רכיב עם מופע Session? בפרק 9 תלמד כיצד להשתמש ברכיב Ad Rotator. רכיב זה משמש להצגת כרזת פרסום שונה בתדירויות משתנות. אם אתה מעוניין להציג כרזת פרסום במספר דפים, הרי זה הגיוני לשייך את הרכיב Ad Rotator למשתנה session.

## יצירת רכיבים עם טווח יישום (Application Scope)

כאשר אתה יוצר מופע עבור רכיב עם טווח יישום (Application Scope) אתה מתייחס אליו כאל אובייקט מובנה. אחרי שהוא נוצר, יכול כל משתמש בכל דף לגשת לכל האוספים, השיטות והמאפיינים של הרכיב. הרכיב נשאר זמין עד שהשרת כבה, או עד שהקובץ Global.asa משתנה, או שהיישום נסגר ויוצא מהזיכרון.

**הערה:**

אם נוצר רכיב עם טווח יישום, השיטות שלו ()OnStartPage ו- ()OnEndPage לא נקראות. זה יכול להיות רלוונטי כאשר אתה משתמש ברכיבים מותאמים.

ניתן ליצור רכיב עם טווח יישום תוך שימוש בשיטות דומות לאלו המשמשות ליצירת רכיב עם Session Scope. קודם כל, תוכל ליצור רכיב עם טווח יישום תוך שימוש בשיטה ()Server.CreateObject. ראה את הדוגמה הבאה:

```
<%
Set Application("AdRot")=Server.CreateObject("MSWC.AdRotator")
%>
```

כאן, משויך מופע הרכיב Ad Rotator למשתנה יישום (Application Variable). תוכל ליצור רכיב בדרך זו מתוך כל אחד מדפי ASP שלך. אבל, בדרך כלל, אתה תיצור מופע עבור רכיב עם טווח יישום רק פעם אחת באירוע Application_OnStart של הקובץ Global.asa. אז הופך האובייקט לזמין עבור כל מבקרי האתר שלך.

לאחר שנוצר עבור הרכיב Ad Rotator מופע עם טווח יישום אתה יכול להשתמש במאפייניו בכל דף ASP. לדוגמה, התסריט הבא קורא למאפיין getAdvertisement של Ad Rotator, כדי להציג כרזת פרסום.

```
<%Application( "adRot" ).getAdvertisement( "schedule.txt" )%>
```

ניתן גם להשתמש בתגית <OBJECT> ליצירת רכיב עם טווח יישום, הנה כך :

```
<OBJECT RUNAT="Server" SCOPE="Application" ID="AdRot"
PROGID="MSWC.AdRotator"></OBJECT>
```

כאשר יוצרים מופע עבור רכיב עם טווח יישום בתגית <OBJECT>, ניתן ליצור את המופע בקובץ Global.asa בלבד. צור את מופע האובייקט מחוץ לתת השגרות Application_OnStart, Application_OnEnd, Session_OnStart ו- Session_OnEnd.

בדוגמה זו נעשה שימוש בארבעה מאפיינים של התגית <OBJECT>. המאפיין SCOPE מציין שלרכיב צריך להיות טווח יישום, ולא Session Scope. המאפיין ID מספק שם עבור הרכיב. המאפיין PROGID מאפשר לשרת לזהות את הרכיב.

**הערה:**

כאשר יוצרים רכיבים עם טווח יישום יש להיזהר לגבי מודל בניית המטלות (Threading Model) של הרכיב. לדוגמה,לא ניתן ליצור מופע של הרכיב Browser Capabilities עם טווח יישום, מפני שהוא משתמש במודל בניית מטלות על-פי דירות (Apartment-Threading Model).

לאחר יצירת מופע עבור אובייקט יישום בקובץ Gllobal.asa באמצעות התגית <OBJECT>, תוכל להשתמש ברכיב בדיוק באותו האופן בו אתה משתמש באובייקט מובנה של דף ASP. אתה יכול לקרוא לכל אחת מהשיטות או המאפיינים של האובייקט בכל דף, מבלי ליצור מופע נוסף שלו.

מתי תצטרך ליצור אובייקט עם טווח יישום? בדוגמת הקוד WhosOn שבפרק 4 תלמד כיצד לעקוב אחר בקשות המשתמש לדפים. מידע זה נשמר במילון הנוצר באמצעות טווח יישום. רכיב זה חייב להיווצר עם טווח יישום, אחרת הוא לא יהיה נגיש לכל משתמש מכל דף.

# איתור וטיפול בתקלות ASP

לפני שתוכל להשתמש בדפי ASP עליך לוודא שהוא מותקן כהלכה במערכת. אם התקנת את IIS (Internet Information Server) בעת התקנת שרת Windows NT, ייתכן והרכיב Active Server Pages אינו מותקן. הרכיב Active Server Pages נכלל בגירסה 3.0 ומעלה של IIS (Windows NT כוללת את גירסה 4.0 ואילו Windows 2000 כוללת את גירסה 5.0). כדי להוריד את הגירסה האחרונה של ASP בקר באתר Microsoft באינטרנט, בחלק הדן בתחום IIS. כתובתו http://www.microsoft.com/iis.

לאחר שהרכיב Active Server Pages מותקן, עליך להגדיר את IIS להשתמש בדפי ASP. הגדר לפחות תיקיה אחת באתר האינטרנט שלך עם ההרשאות להפעלת קובץ ASP. עקוב אחר הצעדים הבאים:

**1.** הפעל את Internet Service Manager מקבוצת היישומים Internet Information Server שבתפריט Start.

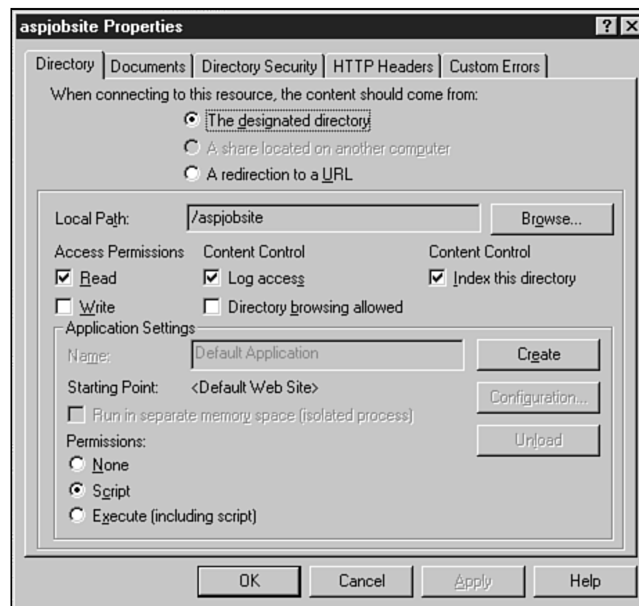
**2.** בחלונית השמאלית של Internet Service Manager, עבור לאתר ברירת המחדל שלך (אם לא שינית את הגדרות ברירת המחדל בהתקנה או לאחריה, ייקרא אתר ברירת המחדל Default Web Site).

**3.** בחר בתיקיה באתר האינטרנט בה אתה מעוניין לשמור את קבצי ASP.

**4.** לחץ לחיצה ימנית על התיקיה הנבחרת ומתפריט הקיצור בחר Properties.

**5.** בחר בכרטיסיה Directory או Virtual Directory (בהתאם לסוג התיקיה הנבחר).

**6.** בחלק Permissions, סמן את האפשרות Script או Execute (ראה תרשים 1.4).




**תרשים 1.4** קביעת הרשאות תיקיה

כעת, משיצרת את התיקיה עם ההרשאות להפעלת דפי ה-ASP שלך, עליך לאחסן את כל דפי ה-ASP שלך בתיקיה זו, כדי שניתן יהיה להשתמש בהם. אם התיקיה היא תיקיה פיסית, אתה יכול לשמור את הקבצים ישירות לתיקיה בעלת שם זה. אם התיקיה היא תיקיה וירטואלית עליך לקבוע נתיב פיסי לתיקיה זו בכונן הדיסק הקשיח שלך. אתה יכול לראות את הנתיב הפיסי לתיקיה הווירטואלית בתיבת הדו-שיח בה הגדרת את ההרשאות (ראה תרשים 1.4).

**אזהרה:**

אם אתה משתמש בדפדפן באותה מכונה בה מותקן IIS, היזהר באופן בו אתה פונה לדפי ASP באמצעות הדפדפן שלך. כאשר אתה טוען דף ASP לדפדפן שלך, אל תטען אותו תוך שימוש בפקודה Open של הדפדפן. השימוש בפקודה זו עוקף את IIS ודף ה-ASP לא יעובד. במקום לראות את מה שאמור התסריט להציג תראה בחלון הדפדפן את התסריט עצמו, וזה לא בדיוק מה שציפית לו.

כדי לטעון דף ASP לדפדפן המותקן באותה מכונה בה פועל IIS השתמש בסרגל הכתובת (Address) של הדפדפן. לדוגמה, אם דף ה-ASP נמצא בתיקיית השורש של אתר האינטרנט שלך, עליך להקליד את הכתובת http://localhost/mypage.asp בסרגל הכתובת של הדפדפן. כתובת זו תיטען לחלון הדפדפן את הדף mypage.asp.

# בדיקת הגדרות

ניתן לבדוק אם הרכיב Active Server Pages מוגדר כהלכה במערכת על ידי יצירת דף ASP פשוט. אם תוכל להציג את דף זה בהצלחה בחלון הדפדפן שלך, אתה יכול להיות בטוח שהכל מוגדר כהלכה.

כדי ליצור דף ASP עליך להיעזר בעורך טקסט פשוט. למשל, תוכל להשתמש בפנקס הרשימות, Notepad, הנכלל בכל מערכות ההפעלה Windows. הפעל את Notepad (או את עורך הטקסט המועדף עליך) והקלד בו את הטקסט הבא:

```
<HTML>
<HEAD><TITLE>ASP Test Page</TITLE></HEAD>
<BODY>
<%
Response.Write(Hello World!")
%>
</BODY>
</HTML>
```

שמור את הקובץ בשם test.asp. ודא שעורך הטקסט שלך אינו מוסיף לו גם את הסיומת txt. בנוסף, ודא שאתה שומר את הקובץ לתיקיה בה הוגדרו הרשאות Execute ו-Script (ראה את הסעיף הקודם).

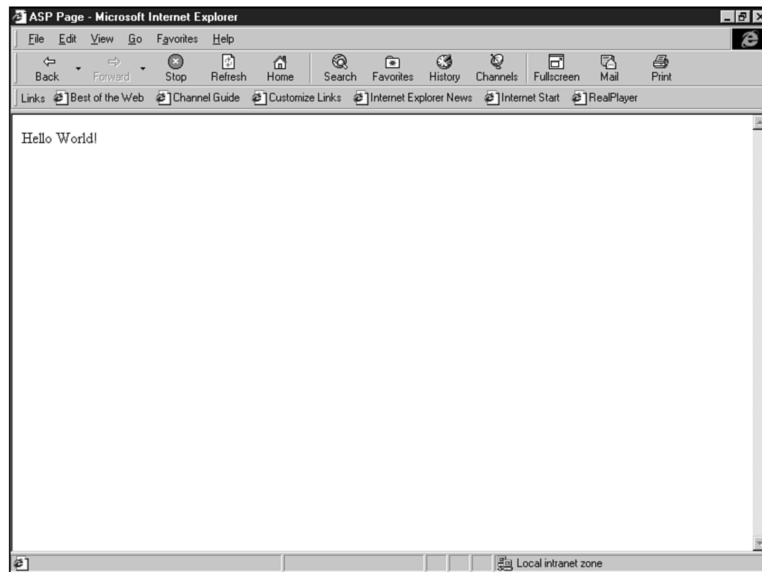
**הערה:**

ניתן למצוא את הקובץ test.asp בתקליטור המצורף לספר זה.
פנה לתיקיה Chap01.

כעת, הפעל את דפדפן האינטרנט המועדף עליך, כדי להציג את דף ה-ASP החדש שיצרת. אם שמרת את הקובץ בתיקיית השורש של אתר האינטרנט שלך, תוכל לטעון אותו על ידי הקלדת הכתובת http://localhost/test.asp. אם שמרת את הקובץ בתיקיית משנה של תיקיית השורש, כלול בכתובת שאתה מקליד את הנתיב המלא למקומו של הקובץ http://localhost/mysubfolder/test.asp. אם שמרת את הקובץ

בתיקיה וירטואלית, כלול בכתובת שאתה מקליד את שם התיקיה הווירטואלית. אם, לדוגמה, קראת לתיקיה הווירטואלית בשם myvirtualfolder, עליך להקליד את הכתובת http://localhost/myvirtualfolder/test.asp.

אם הכל מותקן כהלכה, צריך דף ה-ASP להיות מוצג. הטקסט Hello World! צריך להופיע בחלון הדפדפן (ראה תרשים 1.5). אם קרתה תקלה והדף אינו מוצג, קרא את הסעיף הבא.




**תרשים 1.5** דף ASP

# איתור וטיפול בתקלות הגדרה של דפי ASP

אני מאוד מקווה שלעולם לא תזדקק לקרוא חלק זה. אולם, אם אתה נתקל בבעיות בהצגת דפי ASP, חלק זה אמור לסייע לך. להלן מתוארים תסמינים (Symptoms) אפשריים, והסיבות האפשריות, למספר תקלות אופייניות בהן אתה עלול להיתקל בעת הניסיון לגשת לדף ASP.

- **Symptom** - כאשר מנסים לטעון דף ASP מתקבלת הודעה המתלוננת על כך שהדפדפן אינו יכול להתחבר לשרת. לדוגמה, בעת השימוש ב-Netscape Navigator מופיעה ההודעה:

There was no response. The server could be down or is not responding.

או, בעת השימוש ב-Internet Explorer מתקבלת ההודעה:

Internet Explorer cannot open the internet site http://localhost/test.asp

A connection with the server could not be established.

**סיבה** - אם אינך מנסה להתחבר לשרת דרך האינטרנט, כמעט בוודאות ניתן לומר שבעיה זו נעוצה בעובדה ששרת האינטרנט שלך סגור. ודא ששרת האינטרנט שלך מופעל. כדי לעשות זאת, הפעל את Internet Service Manager (מקבוצת היישומים Microsoft Internet Information Server שבתפריט Start). בחר את שרת ברירת המחדל שלך ובחן את פקדי VCR שבחלקו העליון של החלון. אם IIS אינו פעיל, לחצן VCR Run יהיה לא פעיל (Inactive). לחץ עליו כדי להפעיל את השירות.

**סיבה** - אם אתה מנסה להתחבר לשרת דרך האינטרנט, יכול להיות שהודעת השגיאה נובעת מתנועה ערה (עומס) באינטרנט. המתן מעט ונסה שנית. אם עדיין אינך יכול להתחבר לשרת שלך, צור קשר עם מחלקת התמיכה הטכנית של ספק שירותי האינטרנט שלך.

- **Symptom** - כאשר אתה מנסה לטעון דף ASP מתקבלת ההודעה הבאה:

http/1.0 404 Object Not Found

**סיבה** - הקלדת כתובת שגויה בסרגל הכתובת של הדפדפן. אם שמרת את דף ASP שלך בתיקיית השורש של האתר תוכל לטעון אותו על ידי הקלדת http://localhost/test.asp. אם שמרת את הקובץ בתיקיית משנה של תיקיית השורש, כלול בכתובת שאתה מקליד את הנתיב המלא למקומו של הקובץ, http://localhost/mysubfolder/test.asp. אם שמרת את הקובץ בתיקיה וירטואלית, כלול בכתובת שאתה מקליד את שם התיקיה הווירטואלית. אם, לדוגמה, קראת לתיקיה הווירטואלית בשם myvirtualfolder, עליך להקליד את הכתובת http://localhost/myvirtualfolder/test.asp.

**סיבה** - עורך הטקסט שלך הוסיף סיומת נוספת בסופו של שם קובץ ה-ASP שלך. למשל, כשאתה שומר קבצים אותם אתה עורך באמצעות Notepad, הוא מוסיף להם את הסיומת txt. כדי להמנע מתוספת זו, ודא שבעת שמירת הקובץ נבחרת האפשרות שמירה כסוג (Save as type) כ-All Files. אז לא מתוספת הסיומת.

- **Symptom** - כאשר נעשה ניסיון לטעון דף ASP מתקבלת ההודעה:

http/1.0 403 Access Forbidden. Execute Access Denied.

**סיבה** - ההרשאות שהוגדרו לתיקיה או התיקיה הווירטואלית בה נשמר הקובץ אינן נכונות. קרא שוב את הסעיף הדן בהגדרת הרשאות, מוקדם יותר בחלק זה.

- **Symptom** - כאשר דף ASP נטען מופיע טקסט התסריט, במקום תוצאת התסריט.

**סיבה** - שמרת את קובץ ה-ASP עם סיומת htm או html, במקום asp. כדי ששרת האינטרנט יעבד דפי ASP חייב קובץ התסריט להיות בעל סיומת asp.

**סיבה** - כאשר טענת את הקובץ לדפדפן השתמשת בפקודה Open של הדפדפן. כדי ששרת האינטרנט יוכל לעבד את דף ASP שלך, עליך לטעון אותו תוך שימוש בסרגל הכתובת של הדפדפן.

# סיכום

פרק זה הציג בפניך את השימוש בדפי ASP. למדת את כל השיטות החלופיות להוספת תסריטים לדפי ASP. הפרק גם סיפק סקירה כללית של האובייקטים המובנים ורכיבי ActiveX המוכללים ב-Active Server Pages. לסיום, למדת כיצד להגדיר דפי ASP ולאתר ולטפל בתקלות העלולות להתרחש.

הפרקים הבאים סוקרים בפירוט כיצד להשתמש בכל האובייקטים והרכיבים של ASP. אין זה משנה מה הוא הפרויקט העומד בפניך, הפרקים הבאים יסייעו לך להשיג את התוצאה הדרושה.

## פרק 2

# דפי ASP ופרוטוקול HTTP

**בפרק זה:**

- **השימוש באובייקט Response**
- **השימוש באובייקט Request**
- **שילוב קבצים**

בפרק זה תלמד כיצד להשתמש בשני אובייקטים של ASP הנחשבים בין החשובים ביותר, האובייקט Response והאובייקט Request. שני אובייקטים אלה מספקים לך שליטה מושלמת באופן בו מתַקשֵר שרת האינטרנט שלך עם דפדפני הרשת. אבל, לפני שנתחיל בדיון המעמיק בשני אובייקטים אלה, נדון בקצרה אודות פרוטוקול HTTP, הכוח המניע של ה-World Wide Web, הידוע בקיצור כ-Web או WWW.

כיצד באמת פועל ה-World Wide Web? כאשר אתה מקליד כתובת של דף אינטרנט (Web Page) בדפדפן שלך (Web Browser), אם הכל פועל כשורה, מופיע דף האינטרנט. לדוגמה, אם תקליד http://www.hod-ami.co.il בסרגל הכתובת של הדפדפן שלך, יופיע דף הבית של אתר הוצאת הוד-עמי בחלון הדפדפן. מה קורה ברקע שגורם לכל זה להתרחש?

כשאתה משתמש בדפדפן לאחזור דף HTML מאתר אינטרנט אתה משתמש בפרוטוקול העברת קישורים (**H**yper**T**ext **T**ransfer **P**rotocol, HTTP). פרוטוקול HTTP מגדיר כיצד יכולות הודעות להיות מועברות ברשת האינטרנט. ליתר דיוק, הפרוטוקול קובע את אופן האינטראקציה שבין הדפדפן ושרת האינטרנט.

כשאתה מאחזר קובץ משרת אינטרנט, הדפדפן שלך פותח חיבור (Connection) לשרת אינטרנט שבאתר אינטרנט, ומפיק בקשה (Request). שרת האינטרנט מקבל את הבקשה ומפיק תגובה (Response). מסיבה זו נקרא פרוטוקול HTTP גם בשם פרוטוקול בקשה ותגובה (Request and Response Protocol).

כל התקשורת בין הדפדפן לשרת האינטרנט מתבצעת על ידי צמדי בקשה/תגובה לא רציפים. תמיד זה הדפדפן שאמור ליזום את התקשורת, על ידי הפקת הבקשה. תפקידו של שרת האינטרנט הוא פסיבי לחלוטין; יש לגרום לו לפעול על ידי הגשת הבקשה.

לבקשת דפדפן יש מבנה מסוים. בקשת HTTP מכילה שורת בקשה (Request Line), שדות כותרת (header Fields) וייתכן שגם גוף הודעה (Message Body). הסוג השכיח ביותר של בקשה הוא בקשה פשוטה לדף אינטרנט, כפי הנראה בדוגמה הבאה:

```
GET /hello.htm HTTP/1.1
Host: www.aspsite.com
```

זו דוגמה לבקשת HTTP לדף האינטרנט hello.htm שבאתר האינטרנט www.aspsite.com. השורה הראשונה היא שורת הבקשה. שורת הבקשה מציינת את שיטת הבקשה, את המשאב המבוקש ואת גרסת פרוטוקול HTTP בה נעשה שימוש.

בדוגמה זו, שיטת הבקשה היא השיטה GET. השיטה GET מאחזרת משאב מסוים. במקרה זה, השיטה GET משמשת לאחזור דף אינטרנט בשם hello.htm. שיטות בקשה נוספות כוללות את השיטות POST, HEAD, OPTIONS, DELETE, TRACE ו-PUT. השיטה POST, למשל, משמשת לשליחת תוכן של טופס HTML.

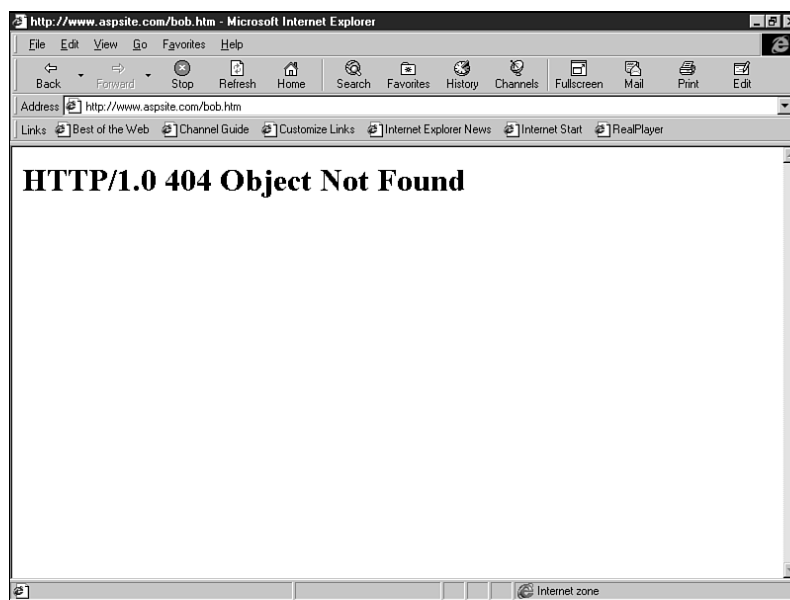
השורה השנייה בדוגמה זו היא כותרת (Header). הכותרת Host מציינת את כתובת האינטרנט של אתר האינטרנט בו מצוי הקובץ hello.htm. במקרה זה, המחשב המארח (host) הוא www.aspsite.com.

בדרך כלל תכיל הבקשה מספר רב של כותרות. כותרות מספקות מידע נוסף אודות תוכן ההודעה או אודות מפיק הבקשה. חלק מכותרות אלו הן סטנדרטיות, אחרות הן ייחודיות לדפדפן בו מופקת הבקשה.

הבקשה יכולה להכיל גם גוף הודעה (Message Body). לדוגמה, אם הבקשה משתמשת בשיטה POST, ולא בשיטה GET, גוף ההודעה יכול להכיל את התוכן של טופס HTML. כאשר אתה לוחץ על לחצן Submit (שלח טופס) בטופס HTML, ובטופס נעשה שימוש במאפיין ACTION="POST", כל מידע שאתה מקליד לטופס נשלח לשרת. תוכן הטופס נשלח בגוף ההודעה של הבקשה, תוך שימוש בשיטה POST.

כאשר שרת אינטרנט מקבל את הבקשה הוא משיב בתגובת HTTP (HTTP Response). גם לתגובה יש מבנה מסוים. כל תגובה מתחילה בשורת מצב (Status Line), המכילה מספר כותרות (Headers) ועשויה לכלול גם גוף הודעה.

קרוב לוודאי שאתה כבר מכיר את שורת המצב. אם אי פעם ביקשת דף אינטרנט ושגית בהקלדת הכתובת, ראית דוגמה לשורת מצב (ראה תרשים 2.1). שורת מצב מציינת את הפרוטוקול בו נעשה שימוש, קוד מצב והודעת טקסט (הנקראת משפט סיבה, Reason Phase). למשל, אם שרת אינטרנט נתקל בבעיה עם בקשה, הוא מגיב בהודעת שגיאה ותיאור שלה בשורת המצב. אם השרת יכול להגיב כהלכה לבקשה לדף אינטרנט, מכילה שורת המצב את הקוד 200 OK.




**תרשים 2.1** דוגמה לשורת מצב

כותרות תגובה מכילות מידע אודות תוכן התגובה, או מידע אודות השרת המספק את התגובה. חלק מהכותרות הללו הן סטנדרטיות, וחלקן תלויות בשרת האינטרנט.

לסיום, גוף ההודעה של תגובה מכיל בדרך כלל את תוכן דף האינטרנט. למשל, אם הבקשה היתה לדף האינטרנט hello.htm, גוף הודעת התגובה יכיל את תוכן דף HTML ששמו hello.htm. אבל, גוף הודעה יכול להכיל סוגים נוספים של תוכן (מסמכי טקסט, מסמכי Word וכדומה).

דפי ASP כוללים שני אובייקטים מובנים המקבילים לתגובות HTTP להודעות הבקשה והתגובה של פרוטוקול העברת היפר-קישורים. האובייקט המובנה Response של ASP הוא המקביל של הודעת התגובה של HTTP ואילו האובייקט Request הוא המקביל של הודעת הבקשה של HTTP. בפרק זה תלמד כיצד להשתמש באובייקטים אלה כדי לשלוט באופן בו מתקשר שרת האינטרנט עם הדפדפנים הפונים אליו.

# האובייקט Response

האובייקט Response של ASP הוא המקביל של תגובת HTTP. השיטות והמאפיינים של אובייקט זה שולטים באופן בו נשלח המידע משרת האינטרנט שלך לדפדפן האינטרנט.

בפרק הקודם הוצגה בפניך השיטה Write של אובייקט זה. שיטה זו שולחת מחרוזת לדפדפן. לדוגמה, התסריט הבא שולח את המחרוזת !Hello World לדפדפן:

```
<%
   Response.Write "Hello World!"
%>
```

לאובייקט Response יש מספר מאפיינים ושיטות יעילות נוספות. בסעיפים הבאים תלמד כיצד להשתמש באובייקט Response כדי לשלוט באופן בו נאגר (Buffered) ומוטמן (Cached) דף אינטרנט, וכיצד ניתן להפנות דף אינטרנט אחד לאחר.

## אגירת פלט הדף

בדרך כלל, כאשר דף ASP מעובד בשרת, נשלח פלט הדף לדפדפן מייד לאחר שכל פקודה בדף מבוצעת. לדוגמה, ראה את דף ASP הבא:

```
<HTML>
<HEAD><TITLE>Buffer Example</TITLE></HEAD>
<BODY>
<%
FOR i=1 TO 500
   Response.Write(i&"<BR>")
NEXT
%>
</BODY>
</HTML>
```

תסריט זה מציג את המספרים 1 עד 500 לאורכו של מסך הדפדפן. פלט הדף נשלח באופן מיידי לדפדפן, לאחר שכל פקודה בדף מבוצעת. אתה יכול לצפות במספרים מופיעים בזמן אמת לאורכו של הדף.

במצבים מסוימים ייתכן ותרצה לאגור (Buffer) את הפלט של דף ASP. כאשר אתה אוגר את הפלט של דף ASP לא נשלח אף חלק מהדף לדפדפן לפני שמסתיים עיבוד הדף בשרת. הרי הגירסה המתוקנת של התסריט הקודם:

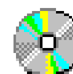
```
<% Response.Buffer=True %>
<HTML>
<HEAD><TITLE>Buffer Example</TITLE></HEAD>
<BODY>
<%
FOR i=1 TO 500
    Response.Write(i&"<BR>")
NEXT
%>
</BODY>
</HTML>
```

קיים רק הבדל אחד בין התסריט הזה לקודם. בשורה הראשונה של תסריט זה נקבע הערך True למאפיין Buffer של האובייקט Response. כאשר דף זה מוצג בדפדפן, נשלח כל תוכן הדף לדפדפן באותו זמן. הדף נאגר (Buffered) עד שמסתיים תהליך עיבוד התסריט.

**טיפ:**

ניתן להשתמש ב- Internet Service Manager כדי לשנות את ערך ברירת המחדל של המאפיין Buffer ל- True. בתיבת הדו-שיח Application Configuration, בכרטיסיה App Options, סמן את האפשרות Enable Buffering.

כל הצהרה המשנה את המאפיין Buffer חייבת להופיע לפני כל קוד HTML או פלט תסריט. אם תנסה לשנות את המאפיין Buffer אחרי קוד HTML או פלט תסריט תתקבל הודעת שגיאה.



מדוע שתרצה לאגור דף ASP? על ידי אגירת הדף אתה יכול להציג שני דפי אינטרנט שונים, בהתאם לתנאים השונים. לדוגמה, דף ASP הבא מציג באופן אקראי שני דפי HTML שונים (תמצא את הקוד בתקליטור המצורף לספר, קובץ בשם buffer.asp):

```
<% Response.Buffer=True %>
<HTML>
<HEAD><TITLE> First Page </TITLE></HEAD>
<BODY>
This is the First Page
</BODY>
</HTML>
<%
Randomize
If INT(2*RND)=1 THEN Response.End
Response.Clear
%>
```

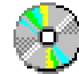
```
<HTML>
<HEAD><TITLE> Second Page </TITLE></HEAD>
<BODY>
This is the Second Page
</BODY>
</HTML>
```

בדוגמה זו נעשה שימוש בשתי שיטות חדשות של האובייקט Response: השיטה End והשיטה Clear. השיטה End עוצרת מייד את תהליך העיבוד של דף ASP ומציגה את התוצאות. אתה יכול להשתמש בשיטה End בין אם אתה אוגר את פלט הדף, ובין אם לא. בדוגמה זו משמשת השיטה End למניעת הצגת הדף השני, כל עוד הדף הראשון מוצג.

השיטה Clear מרוקנת את מאגר הדף הנוכחי מבלי להציג את תוכן המאגר. בשיטה זו ניתן להשתמש אך ורק כאשר אוגרים את הפלט של דף ASP. בדוגמה זו משמשת השיטה Clear למניעת הצגת תוכן הדף הראשון כאשר הדף השני מוצג. היא מנקה את הדף הראשון מהמאגר.

קיימת שיטה נוספת של האובייקט Response, בה משתמשים כאשר אוגרים דף ASP. השיטה Flush מציגה באופן מיידי את תוכן הדף הנאגר. כמו בשיטה Clear, גם כאן תתקבל הודעת שגיאה במידה ותנסה להשתמש בשיטה זו בדף שאינו נאגר. אבל שלא כמו בשיטה End, לאחר שהשיטה Flush נקראת הדף ממשיך להיות מעובד.

בדרך כלל לא תצטרך לאגור את פלט דף ASP. זה אינה פעולה יעילה במיוחד. במקרה של דפי HTML גדולים, או תסריטים ארוכים שלוקח להם זמן להיות מעובדים, אגירת הדף מעקבת את הצגתו, דבר שעלול לבלבל (שלא לומר לתסכל) את המשתמש.



אם אתה מעוניין להציג שני דפי HTML שונים, על פי תנאי כלשהו, תוכל להיעזר בתנאים של VBScript. הנה דוגמת התסריט הקודמת, שנכתבה מחדש ללא אגירה (תמצא את הקוד בתקליטור המצורף לספר, קובץ בשם nobuffer.asp):

```
<%
Randomize
If INT(2*RND)=1 THEN
%>
<HTML>
<HEAD><TITLE> First Page </TITLE></HEAD>
<BODY>
This is the First Page
</BODY>
</HTML>
<% ELSE %>
<HTML>
<HEAD><TITLE> Second Page </TITLE></HEAD>
<BODY>
This is the Second Page
</BODY>
</HTML>
<% END IF %>
```

אחד המצבים הבודדים בהם נחוץ מאגר הוא כאשר אתה משנה את כותרות הדף מתוך גוף הדף. אם אתה מעוניין לשנות את הכותרות בדף לאחר שכבר הצגת את תוכנו בדפדפן, קבע את ערך המאפיין Buffer ל-True. לדוגמה, בסעיף הבא תלמד כיצד להשתמש בכותרות כדי לשלוט באופן בו מוטמן הדף. אם אתה משנה את אחת הכותרות לאחר שכבר הצגת תוכן בדפדפן, עליך לאגור את הדף. אחרת, תתקבל שגיאה.

**הערה:**

אם אתה משתמש בשיטה Response.Redirect (ראה את הסעיף **הפניית משתמש לדף אחר** בהמשך פרק זה), עליך לאגור את הדף במידה ואתה מציג תוכן כלשהו בדפדפן לפני הקריאה לשיטה זו.

# ASP 3.0 Caching

בגירסה 3.0 של ASP אגירת דף ASP פעילה כברירת מחדל, בניגוד לגרסאות קודמות של ASP, בהן היתה ברירת המחדל לאגירה בלתי פעילה.

Microsoft טוענת שהאגירה (Buffering) מהווה אפשרות שליחת דף ASP מהשרת ללקוח באופן יעיל הרבה יותר, וזו הסיבה לכך שברירת המחדל הוחלפה לאגירה אוטומטית ב-IIS 5.0. אגירת ברירת המחדל היא אוטומטית, אלא אם כן שינינו זאת בהגדרות התצורה של ה-IIS, או שהצהרנו בכותרת הדף על כך על ידי הוספת:

```
<% Response.buffer=true %>
```

למעשה, בגירסה 5.0 של IIS רצוי להשתמש באגירה מטעמי יעילות, אך אם קיים דף ASP ארוך, שעיבודו עלול לארוך משך זמן רב מדי, ניתן להשתמש בפונקציות response.flush כדי לשלוח חלקי מידע בכל פעם שהדבר רצוי.

כאשר נשלח אל הלקוח תוכן XML, יש לקבוע, בתחילת דף ה-ASP או על ידי שינוי הגדרות IIS, את הערך FALSE לאגירה. הדבר נעשה כדי לאפשר לדפדפן שמתרגם את תוכן ה-XML לעשות זאת ביעילות ובמהירות, תוך שליחת התוכן מהשרת, ומבלי שהדפדפן יצטרך להמתין עד שיסתיים עיבוד הדף כולו בשרת.

# שליטה במיטמון דפי ASP

ייתכן ואינך יודע זאת, אבל הדפדפן שלך יכול לתעתע בך. כשאתה מבקש דף HTML יכול הדפדפן שלך להחליט שלא לאחזר את הדף מאתר האינטרנט בו מתארח הדף. במקום זאת, הוא עשוי להחליט לאחזר את הדף המבוקש מעותק מוטמן (Cached Copy) השמור בדיסק הקשיח של המחשב שלך.

לדוגמה, כברירת מחדל Internet Explorer יאחזר עותק חדש של דף HTML מהאתר המקורי בו הוא מתארח, רק בכל פעם שהיישום הופעל. לאחר שהדף אוחזר הוא נשמר במטמון הדפדפן. בכל פעם שתבקר ביקור חוזר באותו דף, Internet Explorer ישוב ויציג בפניך את הדף מהמטמון שלו.

דפדפני אינטרנט מטמינים דפים לצורך שיפור ביצועיהם. דף אינטרנט הנטען מהדיסק הקשיח המקומי נטען הרבה יותר מהר מאשר דף אינטרנט המתארח בשרת אינטרנט אי-שם ברחבי רשת האינטרנט.

**הערה:**

ניתן להגדיר כיצד מטמין Internet Explorer את דפי האינטרנט. כדי לעשות זאת פתח את תפריט Tools (כלים), בחר Internet Options (אפשרויות אינטרנט) ובחלק Temporary Internet Files (קבצי אינטרנט זמניים) לחץ על Settings (הגדרות). אם תבחר לבדוק אם קיימת גירסה חדשה יותר של הדף בכל פעם שאתה מבקר בו, אז Internet Explorer יאחזר וישמור עותק חדש של הדף בכל פעם שיימצא כי העותק במטמון ישן יותר מזה שבאתר. שים לב שאין הדבר מחייב שהדפדפן יאחזר עותק חדש של הדף בכל פעם. הוא ימשיך להשתמש בעותק המוטמן של הדף, עד אשר ייאמר לו על ידי שרת האינטרנט שהקובץ המוטמן כבר מיושן.

יכול מאוד להיות גם שהדפדפן שלך אינו היחידי המטמין דפי HTML. שרתי Proxy גם הם משתמשים במטמון לשיפור ביצועים. אם מוצב שרת Proxy בין דפדפן האינטרנט שלך לבין אתר האינטרנט המארח את הדף המבוקש, יכול להיות ששרת ה-Proxy יספק לך עותק מיושן של דף האינטרנט, במקום עותק עדכני.

גם שרתי Proxy וגם מטמון הדפדפן נחשבים למטרד בעיני מתכנתי ASP. כל הייחוד של דף ASP מסתכם בכך שהוא מסוגל לספק תוכן דינמי - תוכן שיכול להשתנות בכל פעם שאתה מאחזר את הדף. אבל, הטמנת דף עלולה לגרום לכך שהדפים שיוצגו בדפדפן לא יהיו עדכניים.

האובייקט Response מכיל שלושה מאפיינים לשליטה בהטמנת דף: Expires, ExpiresAbsolute ו- CacheControl. המאפיינים Expires ו- ExpiresAbsolute משמשים לשליטה באופן בו הדפדפן מטמין דפי אינטרנט. המאפיין CacheControl משמש לשליטה באופן בו מטמינים שרתי Proxy את דפי האינטרנט.

הפעל את הדפדפן שלך ובקש ממנו לאחזר את דף ASP הבא משרת האינטרנט שלך:

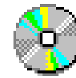
```
<HTML>
<HEAD><TITLE> The Time </TITLE></HEAD>
<BODY>
The Time Is: <%=TIME()%>
</BODY>
</HTML>
```

דף זה מציג את השעה הנוכחית. אם תשוב ותבקש את דף ASP זה על ידי הקלדת כתובתו בסרגל הכתובת של הדפדפן והקשה על Enter פעם אחרי פעם, הדף ישתנה ויציג את השעה הנוכחית (תלוי בדפדפן שלך ולהגדרותיו, הוא עשוי שלא לעשות כן - ולכן חשוב שתקרא את הסעיף הזה).

כדי להורות לדפדפן באופן מפורש לאחזר עותק חדש של דף זה בכל פעם שהוא מבוקש, עליך להשתמש במאפיין Expires או במאפיין ExpiresAbsolute. לדוגמה, תוקף הדף הבא תמיד פג (Expires) לאחר 0 דקות (או במילים אחרות, מייד):

```
<% Response.Expires = 0 %>
<HTML>
<HEAD><TITLE> The Time </TITLE></HEAD>
<BODY>
The Time Is: <%=TIME()%>
</BODY>
</HTML>
```

הערך של Expires מציין בפני הדפדפן כמה זמן, בדקות, עליו לשמור עותק של הדף במטמון שלו. המאפיין ExpiresAbsolute דומה למאפיין Expires, אבל הוא משמש לקביעת תאריך מוחלט בו פג תוקף הדף.



ייתכנו שני מצבים בהם אתה יודע שהדף לא ישתנה לעיתים קרובות. במקרים אלה ייתכן ותרצה להאריך את משך הזמן בו משתמש הדפדפן בעותק המוטמן של הדף. לדוגמה, הדף הבא יישמר במטמון עד לתאריך 2 בינואר 2001 (תמצא את הקוד בתקליטור המצורף לספר, קובץ בשם absoluteTime.asp):

```
<% Response.ExpiresAbsolute = #JAN 2, 2001 00:00:00# %>
<HTML>
<HEAD><TITLE> The Time </TITLE></HEAD>
<BODY>
The Time Is: <%=TIME()%>
</BODY>
</HTML>
```

אם תשוב ותבקש את הדף על ידי הקלדת כתובתו בסרגל הכתובת של הדפדפן פעם אחר פעם, ימשיך הדף להציג את אותה השעה. הדף יישאר כמות שהוא עד ה-2 לינואר 2001 (לחיצה על לחצן רענן, Refresh (או טען מחדש, Reload, ב-Netscape), תגרום בדרך כלל לטעינת עותק חדש ועדכני).

המאפיינים Expires ו- ExpiresAbsolute של האובייקט Response יכולים לשמש לשליטה באופן בו מטמינים הדפדפנים את דפי האינטרנט. כדי לשלוט באופן בו שרתי Proxy מטמינים דפי אינטרנט יש להשתמש במאפיין CacheControl של האובייקט.

המאפיין CacheControl יכול לקבל אחד משני ערכים. כברירת מחדל, נקבע ערכו של המאפיין כ-Private. הגדרה זו מורה לשרתי Proxy שתוכנו של דף ASP זה הוא פרטי של משתמש מסוים, ואין לטמון אותו. מצד שני, אם אתה מעוניין ששרתי PROXY יטמינו את דף ASP שלך, עליך לקבוע את ערך המאפיין CacheControl כ-Public, ראה להלן בדוגמה הבאה:

```
<% Response.CacheControl = "PUBLIC" %>
<HTML>
<HEAD><TITLE> The Time </TITLE></HEAD>
<BODY>
The Time Is: <%=TIME()%>
</BODY>
</HTML>
```

**הערה:**

ניתן להשתמש גם ב- Internet Service Manager לשליטה בתוקפו של תוכן. פתח את תיבת הדו-שיח של מאפייני האתר שלך ובחר בכרטיסיה HTTP Headers. אם תסמן את תיבת הסימון Enable Content Expiration תוכל לקבוע את תוקף הדף.

# שמירה על קשר עם הלקוח

קרה לך כבר פעם שניהלת שיחת טלפון ארוכה, אבל אחרי שהסברת בפירוט ובאריכות תהליך מסוים גילית שהשיחה נותקה? יכול מאוד להיות שביזבזת זמן יקר ומאמצים. השרת שלך יכול להתקל במצב דומה. בדרך כלל, IIS ימשיך לעבד דף ASP אפילו אם הדפדפן שביקש את הדף במקור כבר ניתק ממנו מגע. אם משתמש מבקש דף ואז מחליט לעבור לאתר אחר, שרת האינטרנט שלך לא יבחין בהבדל.

במידה מסוימת, אתה יכול לעקוף בעיה מסוג זה, תוך שימוש במאפיין isClientConnected של האובייקט Response. למאפיין isClientConnected יש ערך TRUE במידה והלקוח עדיין מחובר לאתר, אחרת יקבל את הערך FALSE. אבל, ניתן להשתמש במאפיין זה רק לאחר שהועברה לדפדפן מחרוזת באמצעות השיטה Response.Write. להלן דוגמה:

```
<%
FOR i = 1 TO 1000
    Response.Write i & "<BR>"
    if NOT( Response.isClientConnected ) then exit for
NEXT
%>
```

תסריט זה מציג את המספרים 1 עד 1000 בלולאת FOR...NEXT. אם הלקוח מתנתק בשלב כלשהו בזמן שהשרת מעבד את הלולאה, מתבצעת יציאה מהלולאה.

# שינויים ב-Response.isClientConnected ב-ASP 3.0

בדרך כלל נשתמש ב-isClientConnected כדי לבדוק שהשרת אינו ממשיך בעיבוד דף ה-ASP בזמן שהלקוח התנתק מהאתר, או עבר לדף אחר, וכך מבוזבזים משאבים יקרים בשרת. המאפיין isClientConnected, אשר בודק אם הלקוח עדיין ממתין לדף, שונה בגירסה 3.0 של ASP, כך שמבחינתנו אין אנו מחויבים לשלוח ללקוח מנה כלשהי לפני שאנו יכולים לבדוק אם הוא עדיין ממתין לדף.

במידה ובבדיקה עלה כי הלקוח התנתק, או ביטל את הבקשה על ידי בקשת דף אחר בעוד השרת מעבד את הבקשה שלו, ניתן לסיים את עיבוד הדף באמצעות Response.End. לדוגמה:

```
<%
for I=0 to 1000
  if Response.isClientConnected=true then
      Response.write "שלום" & i
else
      Response.end
  End if
next
%>
```

בדוגמה זו, אשר עובדת על IIS 5.0, ניתן לראות כי במהלך ביצוע הלולאה מתבצעת בדיקה אם הלקוח עדיין ממתין לתוכן, ללא קשר אם השרת שלח תוכן כלשהו או לא.

## עבודה עם קודי מצב

בכל פעם ששרת מגיב לבקשת דפדפן באמצעות תגובת HTTP, השורה הראשונה שהוא שולח היא שורת המצב (Status Line). שורת המצב כוללת קוד מצב בן 3 ספרות ותיאור קוד המצב (נקרא משפט סיבה, Reason Phase). הרשימה הבאה מתארת את חמש מחלקות קודי המצב:

- 1xx - מידע (Informational) - קודי מצב במחלקה זו נועדו לצורך ניסוי ובדיקה.
- 2xx - הצלחה (Successfully) - קודי המצב במחלקה זו משמשים לציון העובדה שהבקשה מולאה בהצלחה. למשל, קוד מצב 200 מציין שדף האינטרנט המבוקש אוחזר בהצלחה.
- 3xx - הפניה (Redirection) - קודי המצב במחלקה זו משמשים לציון העובדה שיש לבצע פעולה נוספת לפני שניתן למלא את הבקשה. למשל, קוד 301 יכול לציין שדף האינטרנט הועבר לצמיתות לכתובת אחרת. במקרה כגון זה ניתן להפנות את הדפדפן כך שבאופן אוטומטי יופנה לכתובת החדשה.
- 4xx - שגיאת לקוח (Client Error) - קוד מצב זה מוחזר כאשר הדפדפן ביקש בקשה שלא ניתן למלא. למשל, קוד מצב 404 מציין שדף האינטרנט המבוקש לא קיים.
- 5xx - שגיאת שרת (Server Error) - קודי מצב במחלקה זו מצביעים על בעיה בשרת. למשל, קוד מצב 503 עשוי לציין שהשרת כרגע עמוס מעבר ליכולתו.

**הערה:**

לקבלת רשימה מלאה של קודי המצב ומשמעותם, קרא את מפרט HTTP 1.1 המפורסם באתר http://www.w3.org.

ניתן להשתמש במאפיין Status של האובייקט Response, כדי לציין קוד מצב שיוחזר בתגובה. לדוגמה, אם מישהו מנסה לאחזר את דף ASP הבא ביום רביעי, יוחזר לו קוד המצב 401 Not Authorized (וכתוצאה מכך גם תוצג תיבת הדו-שיח להקלדת סיסמה):

```
<%
IF WEEKDAYNAME(WEEKDAY(DATE))="Wednesday" THEN
Response.Status="401 Not Authorized"
```

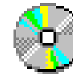
```
Response.End
ELSE
%>
<HTML>
<HEAD><TITLE> Not Wednesday </TITLE></HEAD>
<BODY>
Welcome! Today is NOT Wednesday.
</BODY>
</HTML>
<% END IF %>
```

## שינוי הכותרת Content-Type

הכותרת Content-Type מציינת את סוג המדיה של גוף התגובה (סוג ה-MIME). דוגמאות שכיחות הן text/HTML, image/gif, application/msword או text/rtf. אתה יכול להשתמש במאפיין ContentType של האובייקט Response כדי לקבוע כותרת זו.

**הערה:**

כדי לצפות בסוגי MIME בהם תומך המחשב שלך, פתח את סייר Windows (Windows Explorer), פתח את תפריט View (תצוגה), בחר Folder Options (אפשרויות תיקיה) ובחר בכרטיסיה File Types (סוגי קבצים), לחץ על סוג קובץ כלשהו ובחן את ערך הכותרת MIME Type (סוג תוכן).



שימוש שכיח במאפיין ContentType הוא להצגת מקור HTML של דף אינטרנט. אם תגדיר את ערך המאפיין כ-text/plain יישלח גוף התגובה כטקסט רגיל, ולא כתוכן HTML. ראה את הדוגמה הבאה (הקוד נמצא בתקליטור המצורף לספר, קובץ בשם contentType.asp):

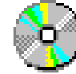
```
<%
Response.ContentType="text/plain"
%>
<HTML>
<HEAD><TITLE> HTML Document </TITLE></HEAD>
<BODY>
<H1>This is an HTML document!</H1>
</BODY>
</HTML>
```

כשקובץ זה יוצג בדפדפן Netscape Navigator יופיע כל הטקסט שמתחת לתסריט בדיוק כפי שהוא מופיע בדף זה. על ידי הגדרת ערך המאפיין ContentType כ-"text/plain" אתה מונע מדפדפן זה מלתרגם את התוכן לתוכן HTML.

לרוע המזל, פעלול זה עובד רק עם דפדפני Netscape. הוא לא יפעל בדפדפני Internet Explorer, מפני שהוא פשוט בוחן את הדף, מאתר תגיות HTML ומחליט כיצד יש לתרגם ולהציג את הדף. אם אתה מעוניין להציג דף טקסט נקי, עליך להסיר את כל תגיות HTML ממנו.

# הפניית משתמש לדף אחר

במצבים מסוימים תצטרך להפנות את המשתמש לדף אחר. לדוגמה, אם משתמש מנסה לגשת לדף הדורש הרשמה, צריך להעביר אותו באופן אוטומטי לדף ההרשמה. או, במקרה והמשתמש לא הקליד את כל הנתונים הנדרשים בטופס ההרשמה הוא או היא צריכים להיות מופנים, באופן אוטומטי, לדף המכיל את טופס ההרשמה, כדי שישלימו את הפרטים החסרים.



הפניית משתמש לדף אחר היא פעולה קלה מאוד, כאשר מדובר בדפי ASP. השיטה Redirect של האובייקט Response מאפשרת לך להפנות משתמש לדף חדש. ראה את הדוגמה הבאה (תמצא את הקוד בתקליטור המצורף לספר, קובץ בשם redirect.asp):

```
<%
IF Request.Form("FirstName")="" THEN Response.Redirect "/register.asp"
%>
<HTML>
<HEAD><TITLE> Registration Results </TITLE></HEAD>
<BODY>
Thank you <%=Request.Form("FirstName")%> for registering!
</BODY>
</HTML>
```

תאר לעצמך שהמשתמש זה אך סיים למלא טופס הרשמה והדף חוזר. השיטה Response.Redirect שבדוגמה זו משמשת להחזרת המשתמש חזרה לדף עם טופס ההרשמה, במידה והמשתמש לא הקליד את שמו הפרטי.

צריך להשתמש בשיטה Response.Redirect לפני שטקסט כלשהו מוצג בדפדפן. בשל כך, הרי זה הגיוני להציב שיטה זו בתסריט המופיע מעל התגית <HTML>. הדרך היחידה לעקוף דרישה זו היא לאגור את הפלט של דף ASP שלך (ראה את הסעיף, **אגירת פלט הדף**, מוקדם יותר בפרק זה, כדי ללמוד כיצד לבצע זאת).

ניתן להשתמש בשיטה Response.Redirect כדי להפנות משתמש לכל כתובת URL חוקית. זה יכול להיות דף אחר באתר שלך, או אפילו דף המאוחסן בשרת אינטרנט אחר, אי-שם ברחבי רשת האינטרנט.

לשיטה Response.Redirect יש את הפוטנציאל להיות יעילה מאוד. Microsoft משתמשת בשיטה זו לעיתים קרובות ביישומים להתנסות לדפי ASP, ובאתר האינטרנט שלה. למרבה הצער, יש גם בעיות עם השיטה הזו.

השיטה Response.Redirect פועלת על ידי החזרת קוד מצב מסוים. כאשר נקראת השיטה Response.Redirect מוחזר קוד מצב 302 Object Moved. כותרת מיקום (Location Header) נוספת גם היא לתגובה, כדי לספק את המיקום החדש של הדף המבוקש. קוד המצב וכותרת המיקום יכולים להפנות את הדפדפן באופן אוטומטי לדף החדש.

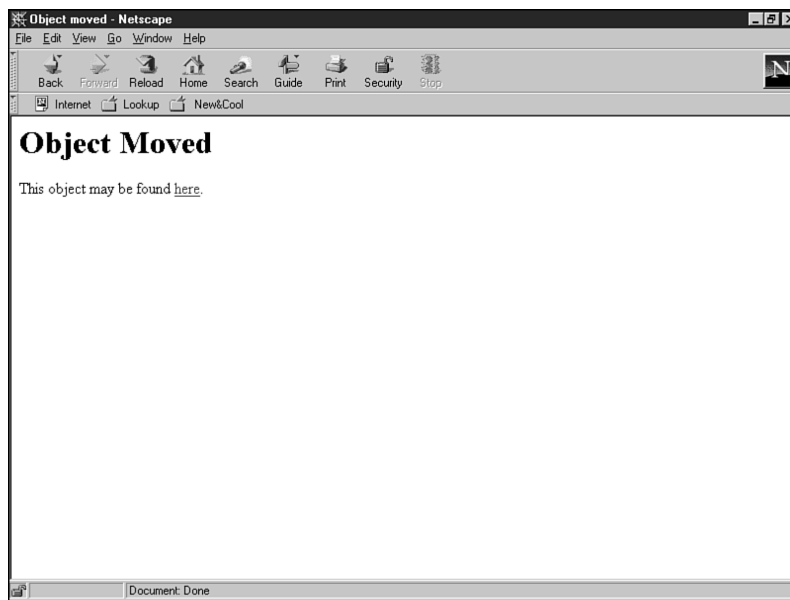
**הערה:**

השיטה Response.Redirect היא המקבילה המדויקת לשתי שורות הקוד הבאות:

```
<%
Response.Status="302 Object Moved"
Response.AddHeader "Location" "URL"
%>
```

למידע נוסף אודות השיטה AddHeader פנה לנספח B של הספר.

לעומת זאת, במציאות, לא תמיד מופנה המשתמש לדף החדש בצורה חלקה כל כך. מי שייתקלו בבעיות בהפניה מסוג זה הם בעיקר דפדפנים מיושנים. וגרוע מכך, אפילו דפדפנים חדשים יחסית, כגון Netscape Navigator 4.0, עשויים להיתקל בבעיה בהיענות אוטומטית לקוד המצב המפנה אותם למיקום אחר. כאשר דפדפן אינו מסוגל להגיב באופן אוטומטי לקוד מצב, תתקבל על המסך הודעה כמו זו בתרשים 2.2.




**תרשים 2.2** תוצאה של הפניית שרת

ההודעה שבתרשים אינה נעימה לעין ועלולה לבלבל את המשתמש. מסיבה זו עליך לנסות ולהימנע מהשימוש בשיטה Response.Redirect. במקום זאת, השתמש בשיטת ההפניה המדומה (Simulated Redirect Method), בה דן הסעיף **הכללת קבצים** שבסופו של פרק זה.

**הערה:**

הגירסה החדשה יותר של ASP, זו המצורפת ל- Windows 2000, כוללת שיטות טובות יותר להפניה אוטומטית של משתמשים לדפים אחרים. כדי ללמוד על כך, קרא את פרק 26.

# בעייתיות בפונקציה Server.redirect

רבים מאתנו לא מבינים שכאשר אנו משתמשים בפונקציה Server.redirect, פעולה זו אינה גורמת לשרת לטעון ולהפעיל דף אחר באופן אוטומטי.

מה שפעולה זו עושה למעשה, זה להוסיף כותרת HTTP (HTTP-HEADER) למחרוזת הפלט הנשלחת מהשרת ללקוח.

כותרת HTTP נראית דומה לזו:

```
HTTP/1.1 302 Object moved
Location newpage.asp
```

קוד המצב הסטנדרטי 302 Object moved, אומר לדפדפן שהדף המבוקש עבר למיקום שונה. הכותרת Location מציינת לאן עבר הדף, כלומר, מאין יש להביא אותו. כאשר הדפדפן מקבל את ההודעה 302 Object moved, הוא מגיב לכך בביטול הבקשה הנוכחית והחלפתה בבקשה לדף החדש, כפי שמצוין בכותרת Location. דבר זה עובד בדיוק כפי שעובדת התגית META HTTP-EQUIV בחלק <HEAD> של הדף.

```
<META HTTP-EQUIV=REFRESH CONTENT=0;URL=newpage.asp>
```

אם כך, למעשה פעולת ההפניה מתבצעת בלקוח, ולא בשרת. למרות שבדרך כלל דבר זה אינו מהווה בעיה, עלולים להיווצר מצבים לא רצויים כאשר קיים שרת Proxy בתמונה. שרת Proxy יירט את הקריאות השונות וייצר בעצמו דף, אשר יישלח ללקוח. זו הסיבה לכך שמדי פעם מופיעה הודעה האומרת שהדף המבוקש הועבר אך ניתן למצוא אותו בקישור הבא (The object you have requested has been moved, and can be found here).

הבעייתיות השנייה היא ביצועים איטיים יותר. היות ובסופו של דבר מתבצעת פעולת ההפניה בלקוח, הרי שבעצם מתבצעת בקשה נוספת לדף אחר, מה שאומר בקשה אחת יותר מדי.

# פתרון בעיית ההפניה ב-ASP 3.0

ב-ASP 3.0 ו-IIS 5.0 אנו יכולים להימנע מבעיית ההפניה בצד הלקוח. כדי לעשות זאת ניעזר ביתרונות המסופקים לנו על ידי שתי פונקציות חדשות Execute ו-Transfer שנוספו לאובייקט Server. פונקציות אלו מעבירות באופן ישיר את השליטה לדף האחר, מבלי לחזור קודם ללקוח, כפי שעושה הפונקציה redirect.

ההבדל בין שתי פונקציות אלו טמון בכך שהפונקציה Execute קוראת לדף אחר בדיוק כפי שאנו קוראים לפונקציה בתסריט שלנו. לאחר שמסתיים עיבוד הדף לו קראה הפונקציה Execute, חוזרת השליטה למקום ממנו היא נקראה.

כאשר משתמשים בפונקציה Transfer, השליטה, למעשה, לא חוזרת לאותו הדף ממנו קראנו לפונקציה זו. במקרה זה סיום העיבוד מתרחש כאשר השרת מגיע לסופו של הדף לו קראה הפונקציה Transfer.

מה שהופך את שתי הפונקציות הללו אף ליעילות יותר זה, שכל ההקשר של דף ASP שהפעיל את אחת מהפונקציות האלו, עובר לדף שנקרא על ידי אחת מהפונקציות הללו. ההקשר של הדף הוא בעצם כל האובייקטים RESPONSE, REQUEST, SESSION, APPLICATION, וכל המשתנים שהוגדרו באותו דף ASP שמתוכו הופעלה הפונקציה.

נקודה מעניינת שיש לשים אליה לב היא שגם אם הפעלנו את הפונקציה Transfer או Execute, וקראנו בעזרתה לדף הנמצא בספריה וירטואלית אחרת, בכל זאת יעבור האובייקט Application גם הוא אל אותו הדף, וישמש שם כאובייקט Application.

התוצאה של כל זה היא שהדפדפן, מבחינתו, מקבל את אותו עמוד מקורי שהוא ביקש, מבלי לדעת שמאחורי הקלעים, בשרת, נקרא דף אחר (אפילו כתובת ה-URL בדפדפן לא משתנה) .

בדוגמה הבאה נראה כיצד פועלת הפונקציה Server.Execute :

קובץ worldfile.asp

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<%
   response.write "world"
%>
</BODY>
</HTML>
```

הפעל קובץ executeWorld.asp

```
<HTML>
<HEAD>
</HEAD>
<BODY>
```

```
<%
    response.write "Hello"
    Server.execute "worldfile.asp"
    response.write "we are here again"
%>
</BODY>
</HTML>
```

אם נריץ את הקובץ ExecuteWorld.asp נקבל את הפלט הבא:

Hello world we are here again

שים לב כי הקוד ממשיך אחרי הנקודה בה קראנו ל-Execute ועד סיומו של דף ASP זה.

בדוגמה הבאה נראה כיצד פועלת הפונקציה Transfer.

קובץ TransferWorld.asp

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<%
    response.write "Hello"
    Server.Transfer "WorldFile.asp"
    response.write "we are here again (maybe not)"
%>
</BODY>
</HTML>
```

כאשר נריץ את הקובץ TransferWorld.asp נראה את שורת הפלט הבאה בדפדפן:

Hello World

כפי שניתן לשים לב, הפונקציה Server.Transfer מעבירה את השליטה לדף ה-ASP המצוין כפרמטר, וכאשר מסתיים עיבודו של דף ASP זה, השליטה לא חוזרת אל הדף הקורא, אלא מסתיימת פעולת העיבוד של הקריאה מהלקוח.

# האובייקט Request

האובייקט Request של ASP מכיל את כל המידע אודות בקשת HTTP המתבצעת לצורך אחזור דף ASP. האובייקט Request כולל שלושה אוספים (collections) המקבילים לחלקי בקשת HTTP : האוסף QueryString, האוסף Form והאוסף ServerVariables.

## QueryString Collection

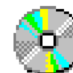
כאשר אתה מכליל מחרוזת שאילתה בבקשה לדף אינטרנט, תוכן מחרוזת השאילתה מאוחזר לתוך האוסף QueryString (QueryString collection). מחרוזת שאילתה היא החלק בבקשה לדף אשר מופיע לאחר הסימן ? (סימן שאלה). לדוגמה, הבקשה הבאה מכילה את מחרוזת השאילתה username=Andrew+Jones :

http://www.aspsite.com/hello.asp?username=Andrew+Jones

מחרוזות שאילתה משמשות להעברת מידע מדפדפן לשרת אינטרנט. בדוגמה זו משמשת מחרוזת השאילתה להעברת משתנה שאילתה בשם username לו יש את הערכים Andrew Jones. מחרוזת השאילתה מועברת לדף בשם hello.asp שבאתר האינטרנט שכתובתו www.aspsite.com.

מתוך דף ASP בשם hello.asp תוכל לאחזר את ערך המשתנה המועבר על ידי מחרוזת השאילתה. התסריט הבא מדגים את הדך להצגת ערך מחרוזת שאילתה בדפדפן :

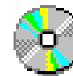
```
<HTML>
<HEAD><TITLE> Query String </TITLE></HEAD>
<BODY>
Welcome <%=Request.QueryString ( "username") %>!
</BODY>
</HTML>
```



בדרך כלל, מחרוזות שאילתה אינן מוקלדות ישירות לסרגל הכתובת של הדפדפן. השימוש במחרוזות שאילתה נעשה בדרך כלל בקישורים, כדי להעביר מידע מדף ASP אחד למשנהו. ראה את הדוגמה הבאה (תמצא את הקוד בתקליטור המצורף לספר, קובץ בשם firstpage.asp) :

```
<HTML>
<HEAD><TITLE>A Choice</TITLE></HEAD>
what is your favorite color?
<P>
<A HREF="nextpage.asp?favcolor=blue">Blue</A>
<P>
<A HREF="nextpage.asp?favcolor=red">Red</A>
</BODY>
</HTML>
```

דף זה מכיל שני קישורים (Links) לדף בשם nextpage.asp. שני הקישורים מכילים משתנה מחרוזת שאילתה בשם favcolor. ערך המשתנה בקישור הראשון הוא blue ואילו ערך המשתנה בקישור השני הוא red.



מתוך דף ASP בשם nextpage.asp (תמצא את הקוד בתקליטור המצורף לספר) אתה יכול לקבוע באיזה משני הקישורים השתמש המבקר באתר כדי לעבור לדף הבא:

```
<HTML>
<HEAD><TITLE>Your Favorite Color</TITLE></HEAD>
<BODY>
<%
 favColor = Request.QueryString( "favColor" )
%>
Your favorite color is <%=favColor%>!
</BODY>
</HTML>
```

דף זה מציג את הצבע המועדף על ידי המשתמש על ידי אחזור המשתנה favColor מהאוסף QueryString (QueryString collection). ערך המשתנה מאוחזר מהאוסף ומשויך למשתנה מקומי, הנקרא גם הוא favColor, ואז המשתנה מוצג.

## קידוד URL של מחרוזות שאילתה

אם ערך משתנה מחרוזת שאילתה מכיל תווים שאינם תווים אלפא-נומריים, כגון מרכאות, רווחים, פסיקים או סימני קריאה, עליך לקודד (Encode) את ערך משתנה מחרוזת השאילתה לפני שניתן יהיה להעביר אותו. לדוגמה, מחרוזת השאילתה הנכללת בקישור הבא לא תוכל להיות מועברת כהלכה בין שני דפים, מפני שהיא מכילה רווח:

```
<A HREF="nextpage.asp?username=Andrew Jones">Next</A>
```

לפני שתוכל להעביר את מחרוזת השאילתה שבדוגמה זו, עליך לקודד אותה קידוד URL. ניתן לעשות זאת באמצעות השיטה URLEncode של האובייקט Server. הרי דוגמה לאופן השימוש בשיטה:

```
<%
   username = Server.URLEncode( "Andrew Jones" )
%>
<A HREF="nextpage.asp?username=<%=username%>">Next</A>
```

בתסריט זה, השיטה URLEncode מחליפה את כל התווים שאינם תווים אלפא-נומריים במחרוזת "Andrew Jones" בתווים הניתנים להעברה במחרוזת שאילתה. לאחר שהמחרוזת קודדה בקידוד URL היא תיראה כך:

Andrew+Jones

הרווח הוחלף בסימן החיבור (+).

דפי ASP אינם כוללים שיטה לפענוח URL של מחרוזת שאילתה, מפני שבדרך כלל לא תצטרך לעשות זאת. כשאתה מאחזר משתנה מחרוזת שאילתה מהאוסף QueryString המשתנה מפוענח פיענוח URL באופן אוטומטי עבורך.

## העברת מספר משתני מחרוזת שאילתה

אינך מוגבל להעברת משתנה מחרוזת שאילתה יחידי במחרוזת שאילתה. כדי להעביר מספר משתנים הפרד בין כל אחד מהם באמצעות הסימן & (Ampersand). מחרוזת השאילתה הבאה משמשת להעברת שני משתנים ששמם username ו-password:

```
<A HREF="nextpage.asp?username=Fred&password=secret">Next</A>
```

מתוך דף ASP בשם nextpage.asp תוכל להתייחס לערכו של כל משתנה על ידי התייחסות לשם המשתנה. התסריט הבא מציג את ערכי שני המשתנים לדפדפן:

```
<%
Response.Write Request.QueryString ( "username" )
Response.Write Request.QueryString ( "password" )
%>
```

אם אתה מעוניין לשייך מספר ערכים לאותו משתנה, אתה יכול להעביר מחרוזת שאילתה כמו זו בדוגמה הבאה:

```
<A HREF="nextpage.asp?color=red&color=blue">Next</A>
```

כאשר מאוחזר משתנה מחרוזת השאילתה ששמו color מהאוסף QueryString יהיו לו שני ערכים, red ו-blue, המופרדים ביניהם בפסיק. לדוגמה, התסריט הבא יציג בדפדפן את המחרוזת "red, blue":

```
<%=Request.QueryString( "color" )%>
```

## QueryStrings עם דפי ברירת מחדל ב- ASP 3.0

אם המשתמש מבקש לראות דף ASP מסוים, אך אינו מציין את שמו המפורש, אלא רושם רק את שם הספריה (למשל /http://www.microsoft.com/scripting) יוצג בפני המשתמש הדף אשר מוגדר כדף ברירת המחדל.

בגרסאות קודמות של ASP, אם לא היינו מציינים במפורש את שם הדף, אלא רק שם תיקיה וירטואלית, ומוסיפים פרמטרים לאחר הכתובת, כמו בדוגמה הבאה:

http://www.microsoft.com/scripting/color=10&page=3

אז דבר זה לא היה מיוחס בשרת.

בגירסה IIS 5.0 יש לכתוב את הדוגמה שלעיל כך :

http://www.microsoft.com/scripting/default.asp?color=10&page=3

וזאת כמובן בתנאי שהקובץ default.asp מוגדר כברירת מחדל עבור התיקיה scripting.

# QueryString Collection

אתה יכול להציג את כל פריטי האוסף QueryString מבלי לדעת את שמם של מי ממשתני מחרוזת השאילתה באוסף. התסריט הבא מבצע איטרציה על האוסף QueryString ומציג כל פריט ופריט, אחד אחרי השני :

```
<%
for i = 1 to Request.QueryString.Count
    Response.Write Request.QueryString( i ) & "<BR>"
next
%>
```

המאפיין Count של האוסף QueryString משמש לאחזור מונה מספר הפריטים באוסף. אם לא הועבר משתנה מחרוזת שאילתה כלשהו, יהיה ערכו של המונה 0.

במקום להשתמש בלולאה FOR...NEXT למעבר על פני האוסף QueryString תוכל להשתמש בלולאה FOR...EACH. היתרון בלולאה זו הוא שאתה יכול להשתמש בה כדי להציג הן את שמו והן את ערכו של כל פריט :

```
<%
for each thing in Request.QueryString
    Response.Write thing & " " & Request.QueryString( thing ) & "<BR>"
next
%>
```

בדוגמה זו, ערך המשתנה thing הוא שמו של כל אחד ממשתני מחרוזת השאילתה באוסף.

**הערה:**

הצגת האוסף Request.QueryString היא פעולה יעילה בעת שמבצעים ניפוי תקלות (Debugging) בדף ASP. אם אינך בטוח אם משתנה מחרוזת שאילתה כלשהו אכן מועבר לדף, הוסף באופן זמני את אחד התסריטים המופיעים כאן לדף שלך.

## מתי אין להשתמש במחרוזות שאילתה

מחרוזות שאילתה יעילות כאשר עליך להעביר פיסות קטנות של מידע מדף אחד למשנהו, אך קיימים שני מצבים בהם ברור שלא תרצה להשתמש בהן : כאשר אתה מעביר מידע חסוי, וכאשר אתה מעביר כמויות גדולות של מידע.

מחרוזות שאילתה אינה נסתרת בשום אופן. מחרוזת השאילתה תופיע תמיד בסרגל הכתובת של הדפדפן. זה אומר, שהעברת סיסמת המשתמש מדף לדף באמצעות

מחרוזת שאילתה היא רעיון רע מאוד. כל מי שיציץ מעבר לכתפו של המשתמש ידע מייד את סיסמתו.

מחרוזות שאילתה הן גם לא הבחירה הנבונה כאשר מדובר בהעברת כמויות גדולות של מידע. המספר המדויק של תווים אותם יכולה מחרוזת שאילתה להכיל תלוי במספר גורמים. אחד מהעיקריים שבהם הוא סוג הדפדפן של המשתמש. לדפדפנים שונים יש מגבלות שונות לגבי אורך מחרוזת השאילתה.

לדוגמה, דפדפן Internet Explorer גירסה 4.0 אינו יכול לטפל במחרוזת שאילתה שאורכה למעלה מ-2000 תווים. קישור עם מחרוזת שאילתה בגודל זה פשוט לא יצליח למלא את תפקידו כקישור (Hyperlink) בדפדפן זה.

אל לך להסיק מנקודה זו שניתן להשתמש במחרוזת שאילתה באורך של עד 2000 תווים. ראשית, האורך המירבי האפשרי תלוי גם באורכה של כתובת ה-URL עצמה. השילוב של כתובת URL ומחרוזת השאילתה (כל מה שמופיע בסרגל הכתובת של הדפדפן) קובע את האורך המירבי.

שנית, קידוד URL של מחרוזת שאילתה גורם בדרך כלל למחרוזת להתארך באופן משמעותי. למשל, נקודות מתורגמות לשלושה תווים (%2E), במקום תו אחד. אי לכך, אורכה המירבי של מחרוזת השאילתה נקבע בהתאם לאורכה של המחרוזת לאחר שעברה קידוד URL.

נקודה שלישית היא שדפדפנים אחרים, לא Internet Explorer, יכולים בדרך כלל לטפל במספר קטן יותר של תווים. כשאורך המחרוזת מגיע ל-1000 תווים בערך, אתה עלול לאבד תאימות עם דפדפנים מסוימים.

בקיצור, רצוי לשמור את מחרוזות השאילתה שלך קצרות ככל האפשר. מחרוזות שאילתה אינן הדרך היעילה ביותר להעברת כמויות גדולות של מידע. וגרוע מכך, דף עם מחרוזת שאילתה ארוכה יכול שלא לתפקד לחלוטין בדפדפנים מסוימים. אם אתה צריך להעביר כמויות גדולות של מידע מדף אחד למשנהו, השתמש בשדה טופס נסתר. דפדפן Netscape, למשל, יכול להעביר שדות טופס המכילים עד 30,000 תווים. פרוטוקול HTTP מעביר שדות טופס באופן יעיל בהרבה מהדרך בה נעשה הדבר במחרוזות שאילתה.

## Form Collection

העבודה עם האוסף Form (Form collection) דומה בעיקרה לעבודה עם האוסף QueryString. האוסף Form משמש לאחסון משתנים המוצבים בטופס HTML. ראה את קוד טופס HTML הבא:

```
<FORM METHOD="post" ACTION="nextpage.asp">
<BR>Username: <INPUT NAME="username" TYPE="text" SIZE="30">
<BR><INPUT TYPE="submit" VALUE="Go!" >
</FORM>
```

טופס זה מכיל שדה טקסט בשם Username ולחצן Submit המציב את הטופס לדף ASP בשם nextpage.asp. מדף זה תוכל לאחזר את הטקסט שהוכנס לטופס:

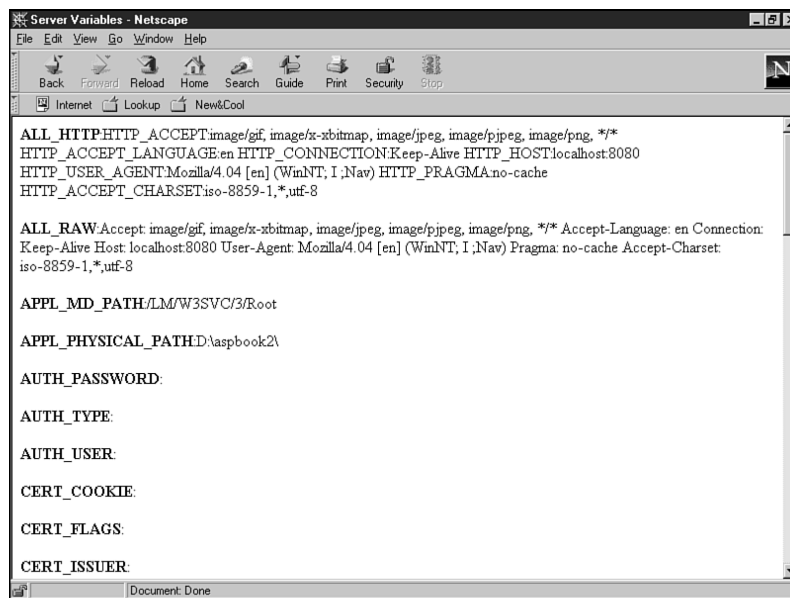
```
<HTML>
<HEAD><TITLE> Next Page </TITLE></HEAD>
<BODY>
<%
   username = RequestForm( "username" )
%>
Your Username is <%=username%>!
</BODY>
</HTML>
```

מכיון שהעבודה עם האוסף Form היא נושא כה חשוב, מוקדש לנושא פרק שלם בספר זה. פרק 3 דן לעומקו של נושא השימוש בשיטות של האוסף Form.

## ServerVariables Collection

כאשר דפדפן מבקש דף אינטרנט משרת, הבקשה כוללת מספר כותרות. אתה יכול לאחזר כותרות אלו תוך שימוש באוסף ServerVariables (ServerVariables collection) של האובייקט Request. האוסף ServerVariables מכיל כותרות ופריטים נוספים של מידע אודות השרת. דף ASP הבא (תמצא את הקוד בתקליטור המצורף לספר, קובץ בשם serverVariables.asp) מעביר את כל תוכן האוסף ServerVariables לחלון הדפדפן:

```
<HTML>
<HEAD><TITLE>Server Variables</TITLE></HEAD>
<BODY>
<%
FOR EACH name IN Request.ServerVariables
Response.write("<P><B>"&name&"</B>:")
Response.write(Request.ServerVariables(name))
NEXT
%>
</BODY>
</HTML>
```




**תרשים 2.3** האוסף ServerVariables

בכותרות (ובמשתני השרת) שבאוסף ServerVariables נשמר מיגוון רחב של סוגי מידע שונים. נספח B מכיל רשימה של הכותרות ומשתני השרת הסטנדרטיים באוסף זה. הרשימה הבאה מסבירה את היעילים יותר מביניהם:

- HTTP_REFERER - כאשר מישהו הופנה לדף זה על ידי לחיצה על קישור, מכילה כותרת זו את כתובת האינטרנט של הדף המפנה. הכותרת HTTP_REFERER היא בעלת ערך רב לקביעת אופן הגעת הגולש לאתר שלך. לדוגמה, אם אתה מעוניין לדעת את מספר הגולשים שהגיעו לאתר שלך מאתר !Yahoo תוכל להיעזר בכותרת HTTP_REFERER כדי לבדוק זאת.
- HTTP_USER_AGENT - כותרת זו מציינת את סוג הדפדפן בו משתמש הגולש באתר שלך. מידע זה חשוב כשאתה מבקש לקבוע באיזה דפדפן משתמש רוב קהל הגולשים באתר שלך.
- REMOTE_ADDR - כותרת זו מכילה את כתובת ה-IP של הגולש באתר שלך. כתובת ה-IP יכולה להיות מתורגמת כך שתספק מידע לגבי מקור המבקרים באתר שלך. למשל, תוכל להיעזר בכותרת זו כדי לקבוע כמה מבקרים באתר שייכים לאיזה ספק שירותים.
- QUERY_STRING - משתנה השרת QUERY_STRING מכיל את החלק של כתובת URL המופיע אחרי סימן השאלה. משתנה זה מכיל את מחרוזת השאילתה השלמה במבנה לא מדולל.
- SCRIPT_NAME - משתנה שרת זה מכיל את הנתיב הווירטואלי לדף ASP הנוכחי. ניתן להשתמש במשתנה זה בדפים המתייחסים לחלקים בתוך עצמם.

- SERVER_NAME - משתנה שרת זה מכיל את כתובת האינטרנט של השרת.
- PATH_TRANSLATE - משתנה שרת זה מכיל את הנתיב הפיסי של דף ASP הנוכחי.

גרסאות מסוימות של Internet Explorer (גירסה 3.0 ו-4.0) כוללות כותרות נוספות בבקשות שלהן:

- HTTP_UA_COLOR - כותרת זו מציינת את מספר הצבעים אותם יכול הדפדפן להציג.
- HTTP_UA_CPU - כותרת זו מציינת את סוג המחשב המשמש להפעלת הדפדפן.
- HTTP_UA_OS - כותרת זו מציינת את מערכת ההפעלה במחשב המשמש להפעלת הדפדפן.
- HTTP_UA_PIXELS - כותרת זו מציינת את רזולוציית המסך המוגדרת במחשב המשמש להפעלת הדפדפן.

**אזהרה:**

אל לך לסמוך על כותרות אלו; הן נתמכות **רק** על ידי גרסאות Internet Explorer הפועלות בסביבת Windows 95.

תוכל לאחזר את תוכנה של כותרת או משתנה שרת מסוים על ידי העברת שמו לאוסף ServerVariables. למשל, התסריט הבא מאפשר למשתמש לגשת לדף רק אם הוא הגיע מדף ASP ששמו origin.asp:

```
<HTML>
<HEAD><TITLE>Server Variables</TITLE></HEAD>
<BODY>
<%
WhereFrom=request.ServerVariables( "HTTP_REFERER" )
If WhereFrom="http://www.mysite.com/origin.asp" THEN
%>
Welcome to this page!
<%
ELSE
%>
You are not authorized to view this page!
<%
END IF
%>
</BODY>
</HTML>
```

בתסריט זה קובעת הכותרת HTTP_REFERER באיזה דף השתמש הגולש כדי לעבור לדף זה. אם הוא לא הגיע מהדף הממוקם בכתובת http://www.mysite.com/origin.asp, לא יינתן לו לראות את תוכן הדף.

**אזהרה:**

שים לב לכך שהשימוש בכותרות באופן זה עלול להיות מסוכן. לא כל הדפדפנים תומכים בכל הכותרות כהלכה. דפדפנים מיושנים רבים אינם תומכים בכותרת HTTP_REFERER.

# הכללת קבצים

אינך יכול להשתמש בפונקציה (function) או שגרה (subroutine) הקיימת בדף ASP אחד, מתוך דף ASP אחר. אם אתה מעוניין ליצור ספריית פונקציות (functions) והליכים (procedures) לשימוש במספר דפי ASP, עליך להציב את הפונקציות וההליכים בקובץ ולכלול אותו בכל דף.

למרבה המזל, ניתן בקלות להוסיף קובץ אחד בדף ASP אחר, תוך שימוש בהכוונת צד-השרת (Server-Side Directive) INCLUDE. ניתן להשתמש בהכוונה INCLUDE כדי לכלול ספריה כללית של פונקציות ושגרות, או לכלול תוכן HTML אחיד במספר מרובה של דפים.

הכוונת INCLUDE של צד-שרת אינה צריכה להופיע בתוך תסריט; השתמש בה מחוץ לתסריט, כחלק מקוד HTML:

```
<HTML>
<HEAD><TITLE> Welcome </TITLE></HEAD>
<BODY>
<!-- #INCLUDE FILE="mybanner.inc" -->
Welcome to our Web Site!
</BODY>
</HTML>
```

בדוגמה זו מוכנס הקובץ mybanner.inc לתוך דף ASP מתחת לתגית <BODY>. כאשר דף ASP זה מעובד, מעובדים יחד איתו גם כל התסריטים וקוד HTML שבקובץ mybanner.inc.

תוכל להכניס קובץ אחד באחר בשתי דרכים: על ידי ציון הנתיב הפיסי לקובץ, כמו בדוגמה הקודמת, או על ידי ציון נתיב וירטואלי לקובץ. הנה דוגמה וירטואלית:

```
<HTML>
<HEAD><TITLE> Welcome </TITLE></HEAD>
<BODY>
<!-- #INCLUDE VIRTUAL="/includes/mybanner.inc" -->
Welcome to our Web Site!
</BODY>
</HTML>
```

אם אתה מספק נתיב פיסי לקובץ המוכלל תוך שימוש בתכונה FILE, חייב הקובץ להיות או בתיקיה הנוכחית, או בתיקיה שהיא תיקיית משנה של התיקיה הנוכחית. הנתיב לקובץ הוא יחסי לתיקיה הנוכחית. כאשר נעשה שימוש בתכונה VIRTUAL, לעומת זאת, יכול הקובץ להיות מאוחסן בכל תיקיה באתר האינטרנט שלך. בדוגמה הקודמת, הקובץ המוכלל מאוחסן בתיקיית המשנה includes/ הממוקמת בתיקיית השורש.

לקובץ המוכלל יכול להיות כל שם וכל סיומת. המוסכמה קובעת שקבצים מוכללים מקבלים את הסיומת inc, אבל אתה יכול להשתמש בסיומת asp, htm, html או כל סיומת אחרת המועדפת עליך.

**הערה:**

אם הקובץ המוכלל מכיל מידע סודי/חסוי, כגון מסד נתוני סיסמאות, עליך ליצור אותו כקובץ ASP על ידי קביעת הסיומת asp עבורו. אם מישהו ינסה לצפות בתוכן הקובץ המוכלל ישירות על ידי הקלדת כתובתו בדפדפן יהיה המידע נסתר, מפני שדף ASP תמיד מעובד לפני שהוא נשלח לתוך הדפדפן.

הכללת קובץ אחד באחר היא פעולה יעילה בשני מצבים. המצב הראשון והמובן מאליו הוא להצגת אותו תוכן או הפעלת אותו תסריט דף אחרי דף. לדוגמה, די נפוץ כיום שכל דף באתר אינטרנט יציג את אותה כותרת עליונה ותחתונה. במקום לשוב על אותו קוד HTML פעם אחר פעם בכל דף, תוכל פשוט לכלול את קובץ הכותרת העליונה והתחתונה כחלק מכל דף.

תוכל גם לכלול את אותו תסריט ASP במספר רב של דפים, תוך שימוש בהכוונת INCLUDE. אבל, מכיון שהכוונת INCLUDE חייבת להתרחש מחוץ לתסריט, חייב התסריט שתכלול להיות כולו מוגבל בין תוחמים (Delimiters). אין אפשרות לכלול רק חלק מתסריט.

המצב השני בו יעיל לכלול קובץ אחד בשני הוא כאשר אתה מעוניין לבצע הדמיה של הפניית שרת (Server Redirection). כדי לעשות זאת אתה מכליל דף ASP שלם אחד בדף ASP שני. ראה את הדוגמה הבאה:

```
<%
IF Request.Form( "FirstName" ) = " " THEN
%>
<!-- #INCLUDE VIRTUAL="/register.asp" -->
<%
Response.End
END IF
%>
<HTML>
<HEAD><TITLE> Registration Results </TITLE></HEAD>
<BODY>
Thank you <%=Request.Form("FirstName")%> for registering!
</BODY>
</HTML>
```

לדוגמה זו יש בדיוק את אותה ההשפעה כאילו השתמשת בשיטה Response.Redirect. אם משתמש שוכח להכניס נתונים לשדה FirstName של טופס ההרשמה הוא יוחזר לדף טופס ההרשמה. אבל, מאחר והכללת הקובץ מתבצעת כולה בשרת, הדמיה זו של הפניה אמינה יותר מאשר הפניה ממשית.

שים לב גם לשימוש בשיטה Response.End שבדוגמה זו. שיטה זו נכללת כדי למנוע מהמשך טופס דף ASP להיות מוצג, אם דף ההרשמה מוצג.

חשוב להבין ש-IIS מעבד את הכוונות INCLUDE לפני שהוא מעבד את התסריטים. דבר זה אומר שאינך יכול להשתמש בפקודת תסריט כדי לכלול באופן דינמי קובץ אחד באחר. לדוגמה, התסריט הבא לא יעבוד:

```
<%
IF Request.Form( "FirstName" ) = " " THEN
    MyInclude="/register.asp"
ELSE
    MyInclude="HomePage.asp"
END IF
%>
<!-- #INCLUDE VIRTUAL="<%=MyInclude%>" -->
```

תסריט זה לא יעבוד מפני שהשרת ינסה לכלול את כל הקבצים קודם להפעלת התסריט. זה אומר שהשרת ינסה לכלול את הקובץ "<%=MyInclude%>", אשר כלל אינו קיים.

## דרך חדשה להכללת קבצים ב-ASP 3.0

במקום להשתמש בתגית <--...#include...--> כדי להכליל קבצים אחרים בדף ASP, ב-ASP 3.0 ניתן להכליל קבצים בהגדרת קוד התסריט עצמו, וזאת באופן הבא:

```
<SCRIPT LANGUAGE="VBSCRIPT" RUNAT="SERVER" SRC="path/filename">
```

## מנגנון אבטחה חזק יותר ב-ASP 3.0

במצבים מסוימים משמשת הכללת קבצים בשרת באמצעות <--...#include...--> להסתרת מידע רגיש, כגון סיסמאות המאפשרות גישה בלעדית למקום כלשהו אליו איננו מעוניינים לאפשר למשתמש להגיע. למשל, כדי להכליל קובץ בו אנו מעוניינים לשמור סיסמה כלשהי, נשתמש בנתיב וירטואלי.

בגרסאות הקודמות של ASP קיימת בעייתיות מסוימת, מפני שאם מישהו יודע את הנתיב הווירטואלי של הקובץ הכלול, הוא יכול לגשת אליו ללא בעיה.

גירסה 3.0 של ASP מציגה פתרון לבעיה זו, על ידי כך שהרשאת הגישה של אותו משתמש המנסה להגיע לקובץ, נבדק אל מול ה-ACL (Access Control List) של אותו הקובץ (רשימת המשתמשים להם מותר לגשת לקובץ הזה) ועל פי זה נקבע האם לאותו משתמש זכות לראות את תוכן הקובץ.

מה שיקרה בפועל זה, שמי שינסה להגיע ישירות מן הדפדפן שלו אל קובץ כלשהו שאין אליו גישה לכל משתמש, ייפתח עבורו חלון המבקש שם וסיסמה. רק אם יסופקו שם משתמש וסיסמה התואמים אחד לשני בקובץ ACL תינתן גישה לקובץ.

# סיכום

בפרק זה למדת כיצד להשתמש באובייקט Response של ASP כדי לשלוט באופן בו מועבר מידע משרת האינטרנט שלך. בנוסף למדת כיצד להשתמש באובייקט Request כדי לאחזר מידע אודות בקשות הדפדפן. לסיום, למדת כיצד לכלול קובץ בדף ASP, תוך שימוש בהכוונת INCLUDE.

בפרק הבא תלמד מספר שיטות מתקדמות לשימוש באוסף Form של האובייקט Request. כך תלמד לשלוט בשיטות העבודה עם טפסי HTML.

## פרק 3

# עבודה עם טפסי HTML

**בפרק זה:**

- **אחזור מידע מטופס**
- **אימות מידע מטופס**
- **השימוש בקובץ ההכוונה Validate Form**
- **קבלת קבצים**

אחד מההיבטים גוזלי הזמן הגדולים ביותר בתהליך יצירת אתר אינטרנט הוא תהליך בניית ועיבוד המידע שייכנס לטפסי HTML. פרק זה יספק לך מספר טריקים וטיפים כיצד להקל על העבודה עם טפסי HTML.

לפרק זה שלוש מטרות. מטרה ראשונה היא לספק לך מספר טכניקות כלליות לשימוש במרכיבים השונים של טפסים בדפי ASP שלך. תיבות טקסט (Text box), אזורי טקסט (Text area), לחצני אפשרויות (Radio button), תיבות סימון (Check box) ותיבות רשימות בחירה (Select List), לכולם יש מאפיינים ייחודיים שעשוי ליצור בעיות ייחודיות כאשר אתה משתמש בהם בדפי ASP שלך. בהמשך יתוארו השיטות לטיפול בכל אחד ממרכיבי הטופס הללו.

חלקו השני של הפרק מוקדש לנושא חשוב והוא אימות של הטופס בצד השרת. כאשר משתמשים מקלידים את נתוניהם לטופס באתר שלך תרצה לוודא שהנתונים אכן הוקלדו כהלכה. בחלקו השני של הפרק תלמד כיצד לבנות קובץ INCLUDE תקני שיבצע את רוב המשימות השכיחות שצריך לוודא.

לסיכום, מרכיב טופס HTML מיוחד ובעייתי יקבל חלק משלו בפרק. בחלק זה של הפרק תתמודד עם השיטות לשימוש במרכיב הטופס File Upload, המאפשר העברת קבצים לאתר שלך. תלמד כיצד לקבל ולהציג קבצי תמונה תוך שימוש בתסריטי ASP תקניים בלבד.

# אחזור נתוני טופס

כאשר משתמש מגיש טופס HTML באתר שלך, כל שדות הטופס וערכיהם ממוקמים באוסף Form (Form collection) של האובייקט Request. מאוסף זה תוכל לאחזר כל שדה טופס על פי שמו, או שתוכל לעבור על פני כל תוכן האוסף Form באמצעות הלולאה FOR...EACH. לדוגמה, ראה את הטופס הפשוט הבא:

```
<HTML>
<HEAD><TITLE>Simple Form</TITLE></HEAD>
<BODY>

<FORM METHOD="post" ACTION="result.asp">
<P>Please Enter Your User Name:
<BR><INPUT NAME="username" TYPE="text">
<P>Please Enter Some Comments:
<BR><TEXTAREA NAME="comments" COLS=40 ROWS=5></TEXTAREA>
<P><INPUT TYPE="submit" VALUE="Save">
</FORM>

</BODY>
</HTML>
```

**הערה:**

האוסף Form כולל את ערכי מרכיבי הטופס כאשר טופס HTML מוגש באמצעות השיטה POST. כאשר טופס HTML מוגש באמצעות השיטה GET ערכי מרכיבי הטופס ממוקמים באוסף QueryString.

טופס HTML זה מכיל שני שדות: המשתמש מתבקש להקליד שם משתמש בתיבת טקסט ולהוסיף הערות באזור טקסט. כאשר המשתמש לוחץ על הלחצן Save, מוגש תוכן הטופס לדף ASP בשם result.asp. מתוך הקובץ result.asp תוכל לאחזר כל אחד משדות הטופס ואת ערכיו, על ידי גישה לאוסף Form.

לדוגמה, תוכל להיעזר בתסריט הבא כדי לאחזר ולהציג את שני מרכיבי הטופס על פי שמם. תסריט זה מאחזר את הפריטים username ו-comments מהאוסף Form, ואז הוא משייך אותם לשני משתני VBScript מקומיים בעלי אותו שם. בשלב זה מוצגים שני המשתנים בדפדפן.

```
<%
DIM username, comment
username = Request.Form( "username" )
comments = Request.Form( "comments" )
%>
<HTML>
<HEAD><TITLE>Result</TITLE></HEAD>
<BODY>
Your User Name Is: <%=username>
<P>
Your Comments Are: <%=comments%>

</BODY>
</HTML>
```

במקום לגשת לכל מרכיב טופס על פי שמו, תוכל גם להציג את כל תוכנו של האוסף Form. התסריט הבא מציג את שמו וערכו של כל מרכיב באוסף Form בחלון הדפדפן, תוך שימוש בלולאה FOR...EACH כדי לפסוע בתוכנו של האוסף Form:

```
<HTML>
<HEAD><TITLE>Simple Form</TITLE></HEAD>
<BODY>
<%
DIM formElement
FOR EACH formElement IN Request.Form
    Response.Write formElement &
    ↳" = " & Request.Form( formElement ) & "<BR>"
NEXT
%>

</BODY>
</HTML>
```

שיטה זו של איטרציה בתוכן האוסף Form יעילה בשני מצבים. ראשית, תמצא אותה יעילה בתהליך ניפוי שגיאות (Debugging) בדפי ASP. אם, מסיבה מסתורית זו או אחרת, אתה נתקל בבעיות באחזור ערך כלשהו של שדה טופס, תוכל להציג את כל תוכן האוסף Form, כדי לקבוע אם שדה הטופס אכן הוגש.

הצגת כל תוכנו של האוסף Form יעילה גם כאשר אינך יודע את שמותיהם של כל שדות הטופס. אם אתה מעוניין לפתח תסריט תקני המסוגל לפעול עם כל טופס HTML, ללא קשר לשמות השדות בו, תוכל להשתמש בשיטה שהוצגה קודם לכן כדי להתעלם לחלוטין ממאפייניו המסוימים של טופס HTML. תשתמש בשיטה זו, המעבר בין כל שדות הטופס, בחלק הבא של פרק זה, כדי לפתח קובץ אימות INCLUDE תקני שיוכל לפעול עם כל טופס HTML.

## שימוש בתיבות טקסט ואזורי טקסט

בחלק הקודם למדת כיצד להיעזר באוסף Form כדי לאחזר את ערכיהם של תיבת טקסט (Text Box) ואזור טקסט (Text Area) וכיצד לשייך ערכים אלה למשתנים. קיימים מספר מצבים בהם עליך לבצע את הפעולה ההפוכה, ולשייך את ערכיו של משתנה לתיבת טקסט (Text Box) או לאזור טקסט (Text Area). לדוגמה, נניח שאתה מעוניין לכלול טופס נוסף, בו יכול המשתמש לאשר את המידע שהוקלד בטופס הראשון. ייתכן ותרצה לאפשר למשתמש הזדמנות נוספת לשנות את הנתונים, לפני שהם מועברים בפועל לבסיס הנתונים. תדפיס 3.1 מציג פעם נוספת, בטופס חדש, את הנתונים שהוקלדו בטופס HTML הקודם.

```
<%
DIM username, comments
username = TRIM( Request.Form( "username" ) )
comments = TRIM( Request.Form( "comments" ) )
%>
<HTML>
<HEAD><TITLE>Simple Form</TITLE></HEAD>
<BODY>

<FORM METHOD="post" ACTION="result.asp">
Do You Really Want To Save This Information?
<P>Your User Name Is:
<BR><INPUT NAME="username"
TYPE="text" VALUE="<%=Server.HTMLEncode( username )%>">
<P>Your Comments Are:
<BR><TEXTAREA NAME="comments" COLS=40 ROWS=5>
   ↳<%=Server.HTMLEncode( comments )%>"></TEXTAREA>

<P><INPUT TYPE="submit" VALUE="Save">
</FORM>

</BODY>
</HTML>
```

טופס HTML זה מציג מחדש את הנתונים שהקליד המשתמש לטופס הראשון. המאפיין VALUE של תגית <INPUT> של HTML משמש לספק את תיבת הטקסט עם ערך. אזור הטקסט מקבל ערך ברירת מחדל על ידי הצבת המשתנה comments בין תגית הפתיחה <TEXTAREA> לתגית הסגירה </TEXTAREA>.

שים לב לשני דברים בטופס זה. ראשית, הפונקציה ()TRIM של VBScript משמשת לחיתוך תווי רווח בתחילת ובסיום הפריטים המאוחזרים מהאוסף Form. בדרך כלל, אם המשתמש אינו מספק ערך עבור תיבת טקסט מוסיף הדפדפן תו רווח עבור השדה הריק כאשר הטופס נשלח. דבר זה יכול להעיק על המשתמש המקליד את הנתונים לטופס. הפונקציה ()TRIM מטפלת בבעיה זו על ידי הסרת תווי רווח לא רצויים אלה.

שים לב שהטופס בתדפיס 3.1 משתמש בשיטה ()HTMLEncode של האובייקט Server. שיטה זו משמשת לקידוד שני המשתנים, לפני שהם יוצגו פעם נוספת בטופס HTML. קיימים תווים מסוימים אותם יכול המשתמש להקליד לטופס HTML אשר עלולים לגרום לטופס שלא להיות מוצג כהלכה. לדוגמה, המאפיין VALUE של התגית <INPUT> משתמש במרכאות כדי לציין את תחילת וסוף המחרוזת. אם למאפיין VALUE משויך משתנה המכיל מרכאות, יוצג המשתנה באופן לא נכון, וטופס HTML יוצג שלא כהלכה. השיטה ()HTMLEncode מטפלת בבעיה זו, על ידי החלפה אוטומטית של סימני מרכאות בתו HTML מיוחד ". הדפדפן, מצידו, מתרגם נכון את התו המיוחד הזה ומציג אותו כסימן מרכאות רגיל.

# שימוש בלחצני אפשרויות ותיבות סימון

לחצני אפשרויות (Radio buttons, הנקראים גם Option buttons) הם דבר יעיל כאשר אתה מעוניין להכריח את ממלא הטופס לבחור מספר מצומצם של בחירות בטופס HTML. לדוגמה, נניח שאתה מעוניין שהמשתמשים יציינו את הצבע המועדף עליהם בעת הרשמה לאתר האינטרנט שלך. אתה יכול להוסיף לטופס HTML שלך שני לחצני אפשרויות עם הכותרות Red ו-Blue, כדי לאפשר למשתמש לבחור בין האפשרויות המוצעות לו:

```
<HTML>
<HEAD><TITLE>Simple HTML Form</TITLE></HEAD>
<BODY>

<FORM METHOD="post" ACTION="result.asp">
<B>Please enter your favorite color: </B>
<BR><INPUT NAME="color" TYPE="radio" VALUE="red"> Red
<BR><INPUT NAME="color" TYPE="radio" VALUE="blue"> Blue
<P>
<INPUT TYPE="submit" VALUE="Save">
</FORM>

</BODY>
</HTML>
```

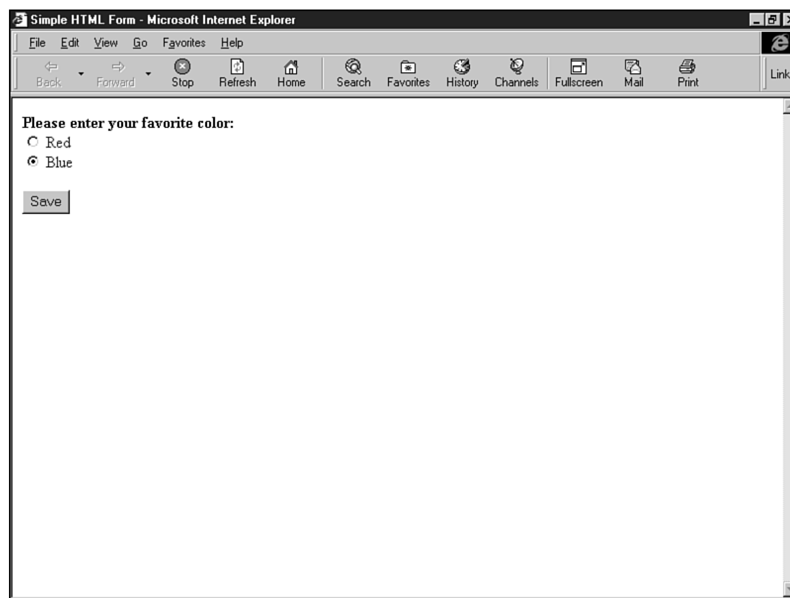
שלא כמו במקרה של תיבת טקסט, אינך יכול להשתמש במאפיין VALUE כדי לספק ללחצן אפשרויות את ערך ברירת המחדל **נבחר** (Checked) או **לא נבחר** (Not Checked). אם אתה מעוניין שערכה של תיבת סימון יהיה תלוי בערכו של משתנה, עליך לכלול פונקציה נוספת בדף ASP שלך. התסריט בתדפיס 3.2 יציג כהלכה סימן נבחר, בהתאם לערכו של המשתנה color.

**תדפיס 3.2** סימון לחצן אפשרויות כנבחר

```
<%
DIM color
color = "blue"

FUNCTION checked( firstVal, secondVal )
    IF firstVal = secondVal THEN
        checked = " CHECKED"
    END IF
END FUNCTION
%>
<HTML>
<HEAD><TITLE>Simple HTML Form</TITLE></HEAD>
<BODY>
```

```
<FORM METHOD="post" ACTION="result.asp">
<B>Please enter your favorite color: </B>
<BR><INPUT NAME="color" TYPE="radio" VALUE="red"
    ↳<%=checked( color, "red")%>> Red
<BR><INPUT NAME="color" TYPE="radio" VALUE="blue"
    ↳<%=checked( color, "blue")%>> Blue

<P><INPUT TYPE="submit" VALUE="Save">
</FORM>

</BODY>
</HTML>
```

מאחר ולמשתנה color יש את הערך Blue, מופיע לחצן האפשרויות השני כנבחר (ראה תרשים 3.1). הפונקציה ()checked משמשת להצגת הסימון. לחצן אפשרויות מופיע כנבחר כאשר התכונה CHECKED מופיעה בתגית <INPUT> של הלחצן. הפונקציה ()checked מקבלת את הערך CHECKED כאשר ערך המשתנה color שווה לערך הארגומנט השני המועבר לפונקציה (הרווח העודף בתחילת הערך המוחזר על ידי הפונקציה, משמש ליצירת רווח ב-HTML).




**תרשים 3.1** טופס עם לחצן אפשרויות

תוכל להשתמש באותה פונקציה גם במקרה של תיבות סימון. כמו לחצני אפשרויות, תיבת סימון מופיעה מסומנת כאשר התכונה CHECKED נכללת בתגית <INPUT>. תדפיס 3.3 מציג דוגמה של טופס עם תיבת סימון העושה שימוש בפונקציה ()checked.

**תדפיס 3.3** סימון תיבת סימון כנבחרת

```
<%
DIM firstvisit
firstvisit = "1"

FUNCTION checked( firstVal, secondVal )
    IF firstVal = secondVal THEN
        checked = " CHECKED"
    END IF
END FUNCTION
%>
<HTML>
<HEAD><TITLE>Simple HTML Form</TITLE></HEAD>
<BODY>

<FORM METHOD="post" ACTION="result.asp">
<B>Check the box if this is your first visit here: </B>
<BR><INPUT NAME="firstvisit" TYPE="checkbox" VALUE="1"
    ↳<%=checked( firstvisit, "1")%>> Red

<P>
<INPUT TYPE="submit" VALUE="Save">
</FORM>

</BODY>
</HTML>
```

# שימוש ברשימות בחירה

רשימות בחירה (Select Lists) מתפקדות בדומה ללחצני אפשרויות. תוכל להיעזר ברשימות בחירה כדי להציג תפריט מוגבל של אפשרויות. אבל, שלא כמו לחצני אפשרויות ותיבות סימון, רשימות בחירה משתמשות בתכונה SELECTED כדי לקבוע איזו אפשרות נבחרת. תדפיס 3.4 הוא דוגמה לטופס עם רשימת בחירה.

```
<%
DIM color
firstvisit = "blue"

FUNCTION checked( firstVal, secondVal )
    IF firstVal = secondVal THEN
        selected = " SELECTED"
    END IF
END FUNCTION
%>
<HTML>
<HEAD><TITLE>Simple HTML Form</TITLE></HEAD>
<BODY>

<FORM METHOD="post" ACTION="result.asp">
<B>Please enter your favorite color: </B>
<SELECT NAME="color">
<BR><OPTION VALUE="red" <%=SELECTED( color, "red" )%>> Red
<BR><OPTION VALUE="blue" <%=SELECTED( color, "blue" )%>> Blue
</SELECT>

<P>
<INPUT TYPE="submit" VALUE="Save">
</FORM>

</BODY>
</HTML>
```

הפונקציה ()SELECTED בדוגמה זו משמשת לבחירת האפשרות התואמת לערך של המשתנה color. מאחר והערך המשתנה color מוגדר כ-blue, האפשרות השנייה נבחרת.

# אימות נתוני טופס

אם אתה משתמש בטופס HTML כדי לאחזר מידע ממשתמש, כמעט תמיד תצטרך לאמת את הנתונים שהמשתמש הקליד. השימושים השכיחים לטפסי HTML הם לאיסוף נתוני רישום משתמש, לביצוע אימות סיסמה או כדי לקבל בקשות להזמנת מוצרים. בניית טופס HTML עצמו היא בדרך כלל הפעולה הקלה ביותר. החלק הקשה הוא למנוע מהמשתמשים להקליד מידע שאינו נכון.

**הערת העורך:**

המונח **Validation** תורגם למילה **אימות** (מהמילה אמת) במשמעות של בדיקת תקינות הנתונים, לבדוק שהנתונים הגיוניים. למשל, תהליך האימות של מחרוזת תווים כלשהי כדי לדעת אם היא מספר טלפון יכלול (בין השאר) חיפוש אחר תווים שאינם ספרות (אסור שבמספר טלפון יימצאו התווים &, % ואחרים). בבדיקה זו לא נבדקת נכונות הנתונים. במקרה של מספר הטלפון: לא נבדק האם הטלפון הזה קיים ו/או המספר מחובר וכדומה. כוונת הבדיקה היא לוודא שהמספר הוא מספר טלפון חוקי

(שאין לו, לדוגמה, רק 5 ספרות) ושהתוכנית בהמשך תוכל להתייחס אליו כמספר טלפון. אין מניעה שתהליך האימות (validation) יכלול בדיקה מקיפה יותר שתבדוק מעבר לבדיקת תקינות בלבד. למשל, אם משתמש הקליד מספר טלפון סלולרי, תהליך האימות יכלול בדיקות למציאת תווים שאינן ספרות (דבר שאסור) וגם יבדוק שהקידומת היא בת 3 ספרות ושתי הספרות שמשמאל הן 05. כי אם הקידומת של הטלפון הסלולרי היא 09 זה לא קידומת של טלפון סלולרי.

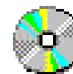
**הערה:**

כדי ללמוד כיצד לשמור מידע מטופס לבסיס נתונים, פנה לחלק 3 של הספר, **שימוש בדפי ASP עם מסדי נתונים**.

בדרך כלל, קיימות שלוש דרישות לאימות נתונים המוקלדים על ידי משתמש לטופס HTML. ראשית, עליך לוודא שכל השדות הנחוצים מולאו כהלכה. אם מישהו מגיש טופס הזמנה באתר האינטרנט שלך, אך שוכח להקליד את מספר כרטיס האשראי שלו, או את הכתובת למשלוח, אתה עשוי לאבד הזמנה.

שנית, רוב האימותים כוללים בדיקה האם סוג הנתונים שהוקלד תואם לסוג השדה. למשל, אם מישהו הקליד את המילה cat בשדה הטופס תאריך לידה (Date of Birth), לא תקבל את המידע לו אתה זקוק.

ושלישית, כאשר נתונים לא נכונים מוקלדים לטופס HTML צריכה להיות מוצגת הודעת שגיאה תואמת. אם המשתמש טעה בכך ששכח למלא שדה כלשהו בטופס, או שהקליד סוג נתון שגוי, עליו לקבל הודעת שגיאה אדיבה המחזירה אותו לטופס המקורי, לביצוע התיקונים הדרושים. כאשר המשתמש מוחזר לטופס המקורי חשוב שהנתונים שכבר הוקלדו לטופס לא יאבדו.



בחלק הבא תלמד כיצב לבנות קובץ INCLUDE תקני לשם אימות נתונים, (Validate Form INCLUDE) אותו תוכל להוסיף לדפי ASP שלך כדי להקל על משימות אימות אלו. קובץ INCLUDE תקני זה יבדוק באופן אוטומטי אם השדות הנדרשים מולאו, האם הוקלד סוג הנתונים הנכון ויציג הודעת שגיאה כאשר הדבר נדרש. הקוד המלא של קובץ INCLUDE זה מופיע בהמשך, וניתן אף למצוא אותו בתקליטור המצורף לספר, קובץ בשם validateform.asp.

# קובץ Validate Form INCLUDE

קובץ Validate Form INCLUDE משתמש בשדות טקסט נסתרים בטופס המוגש כדי לציין איזה שדות הם שדות חובה, ואת סוג הנתונים המתאים להם. על ידי הוספת קובץ Validate Form INCLUDE לדפי ASP שלך והוספת שדות נסתרים אלה לטפסי HTML שלך תוכל לאמת בקלות את טפסי HTML שלך, מבלי להשתמש בתסריטים נוספים.

כדי להדגים כיצד עובד קובץ Validate Form INCLUDE, נתחיל ביצירת טופס HTML פשוט מאוד. טופס זה, כפי שאתה יכול לתאר לעצמך, משמש לאיסוף נתוני משתמש לרישום. הוא מכיל רק שני שדות: שדה עבור שם המשתמש ושדה עבור תאריך הולדתו. הרי הקוד של טופס זה:

```
<HTML>
<HEAD><TITLE>Simple HTML Form</TITLE></HEAD>
<BODY>

<FORM METHOD="post" ACTION="result.asp">
<B>Please enter your name: </B>
<INPUT NAME="username" TYPE="text" SIZE=30>

<P>
<B>Please enter your date of birth: </B>
<INPUT NAME="birthdate" TYPE="text" SIZE=10>
<P>
<INPUT TYPE="submit" VALUE="Save">
</FORM>

</BODY>
</HTML>
```

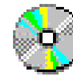
הקלד קוד זה לעורך הטקסט המועדף עליך ושמור את הקובץ בשם form.asp בתיקיה הנגישה על ידי שרת האינטרנט שלך (לדוגמה, wwwroot). תוכל גם להעתיק את הקובץ מהקובץ form.asp שבתיקיה Chapter03 שבתקליטור המצורף לספר זה.

הטופס מוגש לדף ASP בשם result.asp. מטרתו של דף זה היא פשוט להודות למשתמש על הרשמתו לאתר. הקוד של result.asp מופיע בתדפיס 3.5, אותו תוכל להקליד בעצמך, או להעתיק מהקובץ result.asp שבתיקיה Chapter03 שבתקליטור המצורף לספר זה. יש לשמור קובץ זה באותה תיקיה בה שמרת את הקובץ form.asp.

**תדפיס 3.5** הקובץ result.asp

```
<HTML>
<HEAD><TITLE>Result</TITLE></HEAD>
<BODY>

Thank you for registering!

</BODY>
</HTML>
```

אם תפתח את הקובץ form.asp בדפדפן שלך ותלחץ על הלחצן Save, תקבל את ההודעה Thank you for registering!, ללא קשר למידע שהקלדת בטופס. תוכל אפילו להגיש טופס ריק, מבלי להקליד בו מידע כלשהו. אבל, נניח שאינך מעוניין שהמשתמש יגיש לך טופס מבלי שמילא את שני השדות. אם תשתמש בקובץ Validate Form INCLUDE תוכל לקבוע ששני השדות הם שדות הכרחיים.



העתק את הקובץ validateform.asp מהתקליטור (בתיקיה Chapter03) לתיקיה בה שמרת את הקבצים form.asp ו-result.asp. עליך לבצע מספר שינויים בקבצים form.asp ו-result.asp כדי לנצל את יכולותיו של הקובץ validateform.asp.

ראשית, הוסף את השורה הבאה בשורה הראשונה של הקובץ result.asp :

```
<!-- #INCLUDE FILE="validateform.asp" -->
```

הכוונה (Directive) זו של ASP מוסיפה באופן אוטומטי את התוכן של הקובץ validateform.asp לחלקו העליון של קובץ זה.

כעת, עליך לציין איזה שדות טופס הם שדות הכרחיים. מאחר ואיננו מעוניינים שהמשתמש יגיש את הטופס מבלי שהקליד נתונים בשני שדות הטופס, אנו מעוניינים ששני השדות יוגדרו כהכרחיים (Required). הוסף את שני שדות הקלט הנסתרים הבאים לטופס HTML שבקובץ form.asp :

```
<INPUT NAME="username_req" TYPE="hidden"
   ✎VALUE="You must enter a user name.">
<INPUT NAME="birthdate_req" TYPE="hidden"
   ✎VALUE="You must enter your date of birth.">
```

שים לב ששמות השדות הללו זהים לשמות השדות כפי שהופיעו קודם לכן, אך בתוספת _req בסיומם. סיומת זו מסמנת את שני השדות כהכרחיים. התכונה VALUE של שני שדות טופס נסתרים אלה משמשת לאחסון הודעות השגיאה שתוצגנה אם המידע הדרוש לא הוקלד. תוכל לשנות הודעות שגיאה אלו לכל הודעה שתחפוץ.

בשלב הבא עליך להוסיף שדה טופס נסתר נוסף לקובץ form.asp. שדה טופס זה משמש לציון נתיב הקובץ הנוכחי :

```
<INPUT NAME="formScript" TYPE="hidden"
   ✎VALUE="<%=Request.ServerVariables("SCRIPT_NAME")%>">
```

ערכו של משתנה השרת (Server Variable) ששמו SCRIPT_NAME הוא תמיד ערך הנתיב של דף ASP הנוכחי. משתנה זה משמש ליצירת קישור חזרה לטופס, במידה ומתרחשת תקלה.

לסיום, יש להציג מחדש את הנתונים מהטופס המקורי, אם התרחשה תקלה. כדי לשוב ולהציג את שדות הטופס המלאים נוכל להיעזר בשיטות שתוארו בסעיף קודם בפרק זה, **אחזור נתוני טופס**. התכונות VALUE של שני מרכיבי הטופס צריכות להיות משויכות למשתנים המתאימים.

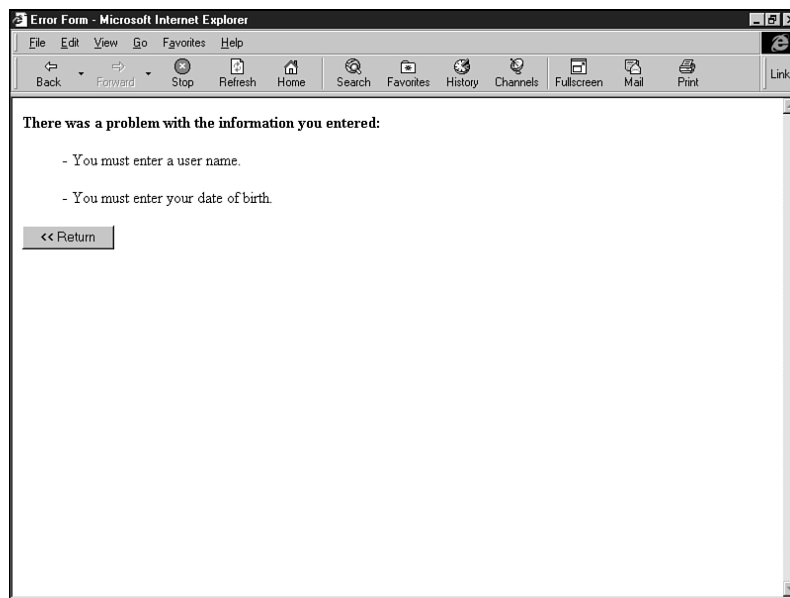
לאחר שביצעת את כל השינויים הללו צריך התסריט המושלם של הדף form.asp להיראות כמו זה המופיע בתדפיס 3.6.

```
<%
DIM username, birthdate
username = TRIM( Request.Form( "username" ) )
birthdate = TRIM( Request.Form( "birthdate" ) )
%>
<HTML>
<HEAD><TITLE>Simple HTML Form</TITLE></HEAD>
<BODY>

<FORM METHOD="post" ACTION="result2.asp">
<INPUT NAME="formScript" TYPE="hidden"
VALUE="<%=Request.ServerVariables("SCRIPT_NAME")%>">

<B>Please enter your name: </B>
<INPUT NAME="username" TYPE="text" SIZE=30
   ➥VALUE="<%=Server.HTMLEncode( username )%>">
<INPUT NAME="username_req" TYPE="hidden"
   ➥VALUE="You must enter a user name.">
<P>
<B>Please enter your date of birth: </B>
<INPUT NAME="birthdate" TYPE="text" SIZE=10
   ➥VALUE="<%=Server.HTMLEncode( birthdate )%>">
<INPUT NAME="birthdate_req" TYPE="hidden"
   ➥VALUE="You must enter your date of birth.">
<P>
<INPUT TYPE="submit" VALUE="Save">
</FORM>

</BODY>
</HTML>
```

שמור את שתי הגרסאות החדשות של הקבצים form.asp ו-result.asp (בתקליטור המצורף לספר, בתיקיה Chapter03 בשמות form2.asp ו-result2.asp) ולחץ על לחצן Save מבלי להקליד בטופס מידע כלשהו. כאשר תלחץ על לחצן זה יופיע בפניך דף הודעת שגיאה המתלונן על כך ששני שדות הטופס אינם ממולאים (ראה תרשים 3.2). אם תלחץ על הלחצן Return יהיה עליך לשוב אל הטופס המקורי.

ההיבט השני שצריך לוודא בטופס הוא בבדיקת האם סוג הנתונים שהוקלד בשדה הוא נכון. הקובץ Validate Form INCLUDE יכול לסייע גם כאן. לדוגמה, שדה תאריך הלידה אמור לקבל רק תאריכים, ולא סוג אחר של מידע. הוסף את שדה הטופס הנסתר הבא לקובץ form.asp:

```
<INPUT NAME="birthday_val" TYPE="hidden" VALUE="date">
```




**תרשים 3.2** טופס הודעות השגיאה

תוכל להוסיף שורה זו בכל מקום שהוא בין התגית <FORM> לבין התגית </FORM> בקובץ form.asp. שדה קלט נסתר זה מגדיר את השדה birthday כשדה תאריך (Date). אם תנסה להגיש את הטופס עם ערך שונה מתאריך, בשדה birthday יופיע דף הודעת השגיאה.

כפי שהוא מופיע בתקליטור, הקובץ Validate Form INCLUDE יכול לבחון שלושה סוגים של נתונים. תוכל להיעזר במילת המפתח date כדי לבדוק תאריך, במילת המפתח number כדי לבדוק ערך מספרי ובמילת המפתח currency כדי לבדוק ערכי מטבע. אבל, אתה יכול לבצע שינויים בקובץ Validate Form INCLUDE כך שיבדוק כל סוג נתונים הרצוי לך. לדוגמה, אתה יכול לבדוק את חוקיותה של כתובת דואר אלקטרוני, מספר זהות או מספרי טלפון. כדי לבדוק סוגים שונים של נתונים פשוט הרחב את ההצהרה SELECT CASE. לדוגמה, ההרחבה הבאה להצהרה SELECT CASE בודקת את חוקיותן של כתובות דואר אלקטרוני:

```
CASE "EMAIL"
IF INSTR( Request.Form( fieldName ), "@" ) = 0
  ➥ OR INSTR( Request.Form( fieldName ), "." ) = 0 THEN
     errorMSG = errorMSG & " - " & fieldName & " must be an E-Mail Address."
END IF
```

הקוד המלא של הקובץ Validate Form INCLUDE מופיע בתדפיס 3.7 להלן.

```
<%
'==================================
'  INCLUDE FILE FOR FORM VALIDATION
'==================================

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'  DECLARE GLOBAL VARIABLES
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
DIM errorMSG

validateForm
IF errorMSG <> "" THEN errorForm

''''''''''''''''''''''''''''''''''''''''''''
'  VALIDATE FORM
''''''''''''''''''''''''''''''''''''''''''''
SUB validateForm
    DIM fieldAttrib, fieldName
        FOR EACH element IN Request.Form
        fieldAttrib   = UCASE( RIGHT( element, 4 ) )
        fieldName = LEFT( element, LEN( element ) - 4 )

        IF fieldAttrib = "_REQ"  AND Request.Form( fieldName ) = "" THEN
            errorMSG = errorMSG & " - " & Request.Form( element ) & "<p>"
        END IF
        IF fieldAttrib = "_VAL" AND Request.Form( fieldName ) <> "" THEN
            SELECT CASE UCASE( Request.Form( element ) )
        CASE "NUMBER"
            IF NOT isNumeric( Request.Form( fieldName ) ) THEN
            errorMSG = errorMSG & " - " & fieldName & " must be a number.<p>"
            END IF
        CASE "DATE"
            IF NOT isDATE( Request.Form( fieldName ) ) THEN
            errorMSG = errorMSG & " - " & fieldName & " must be a date.<p>"
            END IF
        CASE "CURRENCY"
            IF NOT isNumeric( Request.Form( fieldName ) ) THEN
            errorMSG = errorMSG & " - " & fieldName & " must be a money amount.<p>"
            END IF
              END SELECT
        END IF
        NEXT
END SUB
```

```
'''''''''''''''''''''''''''''''''''''''''''''''''''''
' ERROR FORM
'''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB errorFORM
    %>
    <html>
    <head><title>Error Form</title></head>
    <body>
    <b>There was a problem with the information you entered:</b>
    <blockquote>
    <%=errorMSG %>
    </blockquote>
    <form method="post" action="<%=Request( "formScript" )%>">
    <% formFields %>
    <input type="submit" value=" << Return ">
    </form>
    </body>
    </html>
    <%
    Response.End
END SUB

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' DUMP ALL OF THE FORM FIELDS
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB formFields
    DIM element, fieldAttrib
    FOR EACH element IN Request.Form
        fieldAttrib = UCASE( RIGHT( element, 4 ) )
        IF fieldAttrib <> "_REQ" AND fieldAttrib <> "_VAL" THEN
    %>
    <input name="<%=element%>"
    type="hidden"
    value="<%=Server.HTMLEncode( Request.Form( element ) )%>">
    <%
        END IF
    NEXT
END SUB
%>
```

## כיצד פועל הקובץ Validate Form INCLUDE

הקובץ Validate Form INCLUDE פועל על ידי ניצול האוסף Form של האובייקט Request. התסריט שבקובץ INCLUDE עובר על כל אחד ממרכיבי הטופס ובודק אם לכל מרכיב יש את הסיומת _req או val. אם למרכיב יש את הסיומת _req אנו יודעים ששדה הטופס בשם זה, ללא הסיומת _req, הוא שדה הכרחי. אם שדה טופס בשם זה אינו קיים באוסף Form משויכת הודעת שגיאה למשתנה errorMSG.

אם למרכיב כלשהו יש את הסיומת val_ אז אנו יודעים שבשדה הטופס התואם צריך להיות סוג מסוים של נתונים. הצהרת SELECT CASE משמשת כדי לקבוע אם הערך המוקלד בשדה הטופס הוא מסוג הנתונים האמור להופיע בשדה זה. אם לא, נוספת הודעת שגיאה למשתנה errorMSG.

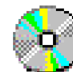
אם הכל עובר כשורה, וכל השדות ההכרחיים מולאו כהלכה בסוג הנתונים הנכון, לא קורה דבר והמשך דף ASP מעובד כרגיל. בדוגמה שהוצגה קודם לכן, תופיע על המסך ההודעה !Thank you for registering.

מצד שני, אם המשתנה errorMSG אינו ריק, אנו יודעים שחסר שדה הכרחי, או שבאחד השדות מופיע סוג נתון שגוי. במקרה שכזה, נקראת השגרה errorForm. השגרה errorForm מציגה את הודעת השגיאה שמכיל המשתנה errorMSG, ואז נעצר תהליך עיבוד התסריט באמצעות ההצהרה Response.End. ההודעה Thank you for !registering לא תוצג לעולם במקרה כגון זה.

כאשר מוצג טופס השגיאה מועבר האוסף Request.Form למשתני טופס נסתרים. הדבר מתבצע כדי שניתן יהיה להעביר חזרה לטופס את כל השדות שמולאו, מה שימנע מהמשתמש את הטרחה שבהקלדה חוזרת של כל הפרטים והנתונים.

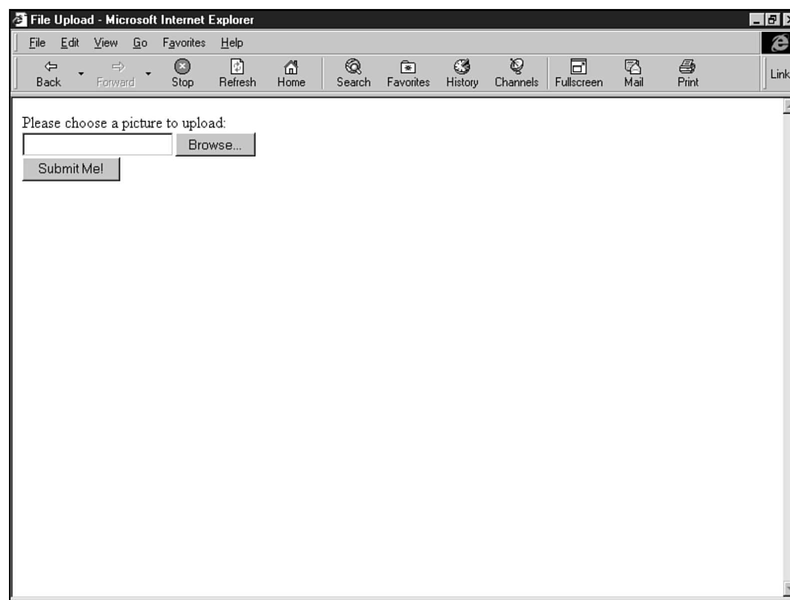
# קבלת קבצים

נניח שאתה מעוניין ליצור אתר אינטרנט בו יכולים משתמשים להציע את ביתם למכירה. יהיה זה נחמד אם המשתמשים יוכלו להעביר גם תמונה של הבית העומד למכירה. ונניח שאתה מעוניין להקים אתר אינטרנט בו יוכלו משתמשים לפרסם סיפורים קצרים מפרי עטם. יהיה זה חביב אם הם יוכלו לעשות זאת על ידי העברת קבצי טקסט, בפורמט של קבצי Word, למשל.



תוכל לעשות זאת באמצעות התכונה TYPE=FILE של התגית <INPUT>. תוך שימוש בתכונה זו תוכל ליצור בטופס שלך לחצן להעברת קבצים (Upload). כאשר מבקר באתר לוחץ על לחצן זה (ראה תרשים 3.3) הוא יכול לבחור קובץ להעברה מהדיסק הקשיח המקומי שלו. הקוד הבא מציג את האפשרות הזו (תמצא את הקוד בתקליטור המצורף לספר, קובץ בשם uploadform.asp):

```
<HTML>
<HEAD><TITLE> File Upload </TITLE></HEAD>
<BODY>

<FORM ENCTYPE="multipart/form-data"
ACTION="uploadresult.asp" METHOD=POST>

Please choose a picture to upload:
<BR><INPUT NAME="picture" TYPE=FILE ACCEPT="image/*">
<BR><INPUT TYPE=SUBMIT VALUE="Submit Me!">
</FORM>

</BODY>
</HTML>
```




**תרשים 3.3** טופס עם לחצן להעברת קבצים

כאשר דף זה מוצג בדפדפן Netscape (גירסה 3 ומעלה) או בדפדפן Internet Explorer (גירסה 4 ומעלה) מופיעה תיבת טקסט רגילה למראה, ולידה לחצן Browse (במערכות הפעלה בעברית הלחצן ייקרא **עיון**). המשתמש יכול להקליד את שם הקובץ ישירות לתיבת הטקסט, או להיעזר בלחצן Browse ולבחור קובץ באמצעות תיבת דו-שיח Choose File (בחר קובץ) או File Upload (תלוי בדפדפן ובמערכת ההפעלה). כאשר הטופס נשלח, נשלחים גם הקבצים שנבחרו על ידי המשתמש.

שים לב בדוגמה זו לתוספת של התכונה ENCTYPE לתגית <FORM>. התכונה ENCTYPE מציינת את סוג הקידוד (Encoding) אותו יש להחיל על נתוני הטופס כאשר הם נשלחים. בדרך כלל, נתוני טופס מקודדים בקידוד URL (URL-Encoded), בו מוחלפים תווי רווח בסימן חיבור (+) וכדומה. אבל, זו שיטת קידוד גרועה לשימוש בסוג נתונים שאינו טקסט. כאשר מעבירים קבצים, עליך להשתמש בתכונה ENCTYPE="multipart/form-date".

לחצן העברת הקבצים נוצר, אם כן, על ידי השימוש בתגית הבאה:

<INPUT NAME="picture" TYPE=FILE ACCEPT="image/*">

מכיון שערך התכונה TYPE הוא FILE על הדפדפן ליצור לחצן להעברת קבצים. התכונה ACCEPT מגבילה את הקבצים שיופיעו בתיבת הדו-שיח Choose File לסוג קובץ מסוים. במקרה שלפנינו מוגבל סוג הקובץ להיות קובץ תמונה (Image). כערך לתכונה ACCEPT אתה יכול לציין כל אחד מסוגי MIME.

**הערה:**

המונח MIME הוא קיצור של Multipurpose Internet Mail Extention. MIME פותח במקור כאמצעי לציון סוגי הקבצים שניתן לצרף להודעות דואר אלקטרוני. דפדפנים משתמשים בסוגי MIME לשיוך קבצים ליישומים התואמים להם.

דוגמאות לסוגי MIME הן: image/gif עבור תמונות בפורמט GIF, image/jpg עבור תמונות בפורמט JPG, application/x-msexcel עבור גיליונות אלקטרוניים של Excel או application/msword עבור מסמכי Word.

כדי לצפות בסוגי MIME הנתמכים על ידי המחשב שלך, פתח את סייר Windows, פתח את תפריט View (תצוגה), בחר Folder Options (אפשרויות תיקיה) ובחר בכרטיסיה File Types (סוגי קבצים).

לחצנים להעברת קבצים עשוים להיות יעילים מאוד. למרבה הצער, דפדפנים רבים תומכים בלחצני העברת קבצים רק באופן חלקי, או כלל לא. לדוגמה, Netscape (גירסה 3.0 או 4.0) מתעלם לחלוטין מהתכונה ACCEPT. Internet Explorer גירסה 3.0 גרוע אף יותר. הוא נכשל לחלוטין בתרגום לחצני העברת קבצים, ובמקום זאת הוא מציג תיבות טקסט רגילות. Internet Explorer גירסה 4.0 או 5.0, לעומת זאת, מזהה ומתרגם כיאות את לחצני העברת הקבצים. אי לכך, השתמש בלחצני העברת קבצים בזהירות הראויה.

# שימוש בדפי ASP לאחזור קבצים מועברים

יצירת טופס HTML עם לחצן העברת קבצים הוא רק הצעד הראשון בקבלת העברת קבצים. אם אתה מגיש קובץ באמצעות לחצן העברת קבצים, אינך יכול להשתמש בשיטות הרגילות של האובייקט Request כדי לאחזר את תוכן הקובץ. האובייקט Request יוצא מנקודת הנחה שטופס נשלח בקידוד ברירת המחדל, URL-Encoded, אבל טפסים המעבירים קבצים אינם משתמשים בסוג קידוד זה.

כדי לקבל קבצים מועברים בתוך דפי ASP, עליך לעבוד עם נתונים גולמיים (Raw Data) המועברים בהצבת HTTP (HTTP Post). למרבה המזל, דפי ASP כוללים שיטה מיוחדת של האובייקט Request במיוחד למטרה זו. השיטה BinaryRead() של האובייקט Request יכולה לשמש לקריאת הבתים הגולמיים (Raw Bytes) המועברים כאשר הטופס נשלח. אולם עליך להיזהר, מפני שאחרי שאתה משתמש בשיטה BinaryRead() בדף ASP אינך יכול יותר להשתמש בשיטה הרגילה Request.Form() לאחזור נתוני טופס (או להיפך).

נניח שמשתמש כלשהו מבקש להעביר תמונה בשם myhouse.gif, תוך שימוש בטופס שתואר בחלק הקודם של פרק זה. התסריט שבתדפיס 3.8 יכול לשמש לאחזור ולהצגת הבתים הגולמיים המועברים באמצעות השיטה BinaryRead().

```
<HTML>
<HEAD><TITLE>Result</TITLE></HEAD>
<BODY>
<PRE>
<%
FormSize = Request.TotalBytes
FormData = Request.BinaryRead( FormSize )
Response.BinaryWrite FormData
%>
</PRE>
</BODY>
</HTML>
```

תסריט זה משתמש במאפיין TotalBytes של האובייקט Request כדי לאחזר את מספר הבתים הכללי שנשלח בהצבת HTTP. אחר כך, כל נתוני הטופס משויכים למשתנה בשם FormData. לסיום, השיטה BinaryWrite() של האובייקט Response משמשת להצגת התוכן של הנתונים המועברים. התוצאה של העברת קובץ תמונה קטן תיראה דומה לזה:

```
----------------------------------- (7),01444'  9=82<. 342ÿÛC
Content-Disposition: form-data; name="picture"; filename="D:\myhouse.gif"
Content-Type: image/gif

GIF89a"
 ¢ יֹש   22222222222222222222222222222222222222222222222!ר!2ףrיֹוְרִיֹוְ "רִוֹ
¦—ו ש  רִוֹ רְטַרְט  רִוֹ וְרִרְלֹל  פוֹ רְרִוֹ וֹרְרִוֹלֹרְלשְׁרִשְׁוֹ ivטLtI<טה œ  µ_‡®&Špl(ˆ-□ÚN6
¾£□½  וי  z_ר%S£Y{□4חמ^Ej{üט¬ויOפפו ל´א □□CK   ¬"qªûv,w  n>vßS
"C¿®□AÛ›(œ□LBû«0ר̇כul Yא \ˆT¯m□N -«לך;0 0י¹ פש`U3ר
```

לא קשה להבחין בכך שפלט זה אינו קריא. אבל, הוא מכיל את כל הנתונים הגולמיים של הקובץ, ומידע חשוב אודות תוכנו. לדוגמה, כותרת השם מכילה את שם שדה הטופס ששימש להעברת הקובץ, כותרת שם הקובץ מכילה את הנתיב לקובץ בדיסק שבמחשב הלקוח והמאפיין content-type מזהה את הקובץ כקובץ תמונה. הנתונים הגולמיים של התמונה מתחילים בשורה המתחילה ב-GIF89a.

לאחר שהנתונים בידך, אתה יכול לשמור אותם לבסיס נתונים, לקובץ או פשוט להציג אותם. אם אתה מרגיש שאפתן במיוחד, תוכל גם לשנות את הדמות הגולמית וליצור דמות מותאמת אישית On The Fly (תוך כדי התהליך). אנחנו, לעומת זאת, עומדים לבצע פעולה פשוטה במיוחד. התסריט הבא (תמצא את הקוד בתקליטור המצורף לספר, קובץ בשם uploadresult.asp) מסיר את הכותרות ומציג את הקובץ שהועבר בחלון הדפדפן:

```
<%
FormSize = Request.TotalBytes
FormData = Request.BinaryRead( FormSize )
bnCRLF = chrB( 13 ) & chrB( 10 )
Divider = LEFTB( FormData,  INSTRB( FormData, bnCRLF ) - 1 )
```

```
DataStart = INSTRB( FormData, bnCRLF & bnCRLF ) + 4
DataEnd = INSTRB( DataStart + 1, FormData, divider ) - DataStart
Response.ContentType = "image/gif"
Response.BinaryWrite MIDB( FormData, DataStart, DataEnd )
%>
```

תסריט זה מחלץ את נתוני התמונה מנתוני הטופס באמצעות הפונקציה ()INSTRB של VBScript, כדי לאתר את תחילת וסיום נתוני התמונה. הוא יוצא מנקודת הנחה שנתוני התמונה מתחילים לאחר שני תווי החזרה (Carriage Return) והזנת שורה (Line Feed) ברצף, וכי נתוני התמונה מסתיימים באותו מפריד (Devider) כמו זה המופיע בתחילת הנתונים. אחר כך, השיטה ContentType של האובייקט Response משמשת לציין שהתוכן שנשלח לדפדפן הוא תמונה בפורמט GIF. לסיום, השיטה ()BinaryWrite משמשת להצגת הנתונים הגולמיים של קובץ התמונה. תמונה זו מוצגת בחלון הדפדפן.

התסריט מניח שהטופס שנשלח מכיל שדה אחד בלבד, וכי שדה זה מכיל נתוני תמונה. אבל, טופס HTML המכיל מרכיב העברת קובץ יכול גם להכיל מרכיבי טופס רגילים נוספים, כגון תיבות טקסט ואזורי טקסט. טופס HTML יחיד יכול אף לכלול מספר מרכיבי העברת קבצים. לדוגמה, להלן הנתונים הגולמיים שנשלחו על ידי טופס יחיד המכיל גם מרכיב העברת קובץ בשם picture וגם אזור טקסט בשם comments:

```
----------------------------------- (7),01444'   9=8 2<. 342ÿÛC
Content-Disposition: form-data; name="picture"; filename="D:\myhouse.gif"
Content-Type: image/gif

GIF89a"
 ¢ יּשׁ   2222222222222222222222222222222222222222222222222!רּ!2רףrיּוּרף "וּרף
¦—ש וּ טּרףטּרף וּרף לּלּרףוּרף וּרףוּרף פּוּ וּשּׁרףלּשּׁרףלּ וּוּרףוּרףLtI<טּi הטּœ  µ_‚‡®&Špl(ˆ-□ÚN6
¾£□½  וי  z_רּ%S£Y{□4חמ^Ej{üד¬ויOפּפּוּ ל´א □□CK‚‚   ¬qªûv,w  n>vßS
″C¿®□AÛ›(œ□LBû«0רּכּul Yא \ˆT¯m□N -«לך;0 0יּ¹ פּשׁ`U3רּ
----------------------------------- (7),01444'9=82<.342ŷ Ci
Content-Disposition: form-data; name="comments"
This is a picture of my house.
----------------------------------- (7),01444'9=82<.342ŷ Ci
```

למרות שניתן לעבוד עם טפסים מורכבים כגון אלה, גם בדפי ASP רגילים, דרך פעולה טובה יותר תהיה להיעזר ברכיב ASP של צד-שלישי. בשימוש ברכיב ASP של צד-שלישי תוכל לעבוד עם טפסים המכילים מרכיבי העברת קבצים ממש כמו עם טפסי HTML רגילים. בפרק 16 תלמד כיצד להשתמש באחד מרכיבים אלה כדי לשלוט בכל היבטי העברת קבצים.

**הערה:**

למידע נוסף אודות השימוש בהעברת קבצים - לך ישר למקור. קרא את המאמר Form-Based File Upload in HTML שנכתב על ידי E. Nebel ו- L. Masinter ואשר נמצא בכתובת http://www.w3.org/MarkUp/Bibliography.html

# סיכום

בפרק זה למדת כיצד להשתמש בדפי ASP לאחזור ולטיפול במידע המוקלד לטפסי HTML. בחלקו הראשון של הפרק למדת כיצד להשתמש במספר טכניקות מתקדמות לעבודה עם תיבות טקסט, אזורי טקסט, לחצני אפשרויות, תיבות סימון ורשימות בחירה. בחלקו השני של הפרק למדת כיצד לבנות קובץ INCLUDE תקני שיכול לשמש לאימות נתוני טופס. בחלקו השלישי והמסיים של פרק זה למדת את רזי השימוש במרכיבי הטופס להעברת קבצים אשר יאפשרו למבקרים באתר שלך להעביר אליו קבצים.

## פרק 4

# עבודה עם יישומי ASP

**בפרק זה:**

- **מהו יישום**
- **יישומים והקובץ Global.asa**
- **שימוש באובייקט Application**
- **דף Chat**
- **דף WhosOn**

בפרק זה תלמד כיצד לעבוד עם יישומי ASP. החלק הראשון סוקר באופן כללי יישומים. בחלק השני תלמד כיצד להשתמש בשיטות ובאוספים של האובייקט Application. לסיום, פרק זה כולל שתי דוגמאות תכנות: תלמד כיצד ליצור תוכנת Chat למספר משתמשים, ודף ASP אשר מציג נתוני זמן אמת לגבי סטטיסטיקת השימוש באתר האינטרנט שלך.

# מהו יישום

Microsoft מעוניינת שתחשוב על דפי ASP במונחי תכנות מקובלים. כשאתה יוצר דף ASP בודד אתה יוצר משהו הדומה לפרוצדורה (procedure) או שגרה (subroutine). כשאתה יוצר קבוצה של דפי ASP אתה יוצר יישום (application).

אבל, יישום הוא מעט יותר מאשר קבוצת דפי ASP המאוחסנים בדיסק קשיח. כאשר דפי ASP מאוגדים יחדיו ליישום, יש להם מספר מאפיינים שבמצבים אחרים יחסרו להם. לפניך רשימה של חלק מהתכונות של יישום דפי ASP:

- ניתן לשתף נתונים בין דפי היישום, ולכן גם בין יותר ממשתמש אחד באתר האינטרנט.
- ליישום יש אירועים אשר יכולים להפעיל תסריטי יישום מיוחדים.
- מופע של אובייקט יכול להיות משותף בין כל הדפים ביישום.
- יישומים נפרדים יכולים להיות מוגדרים על ידי Internet Service Manager כך שיהיו להם מאפיינים שונים.
- ניתן לבודד כל יישום, כך שיפעל במרחב זיכרון משל עצמו. דבר זה אומר שאם יישום אחד קורס, לא יקרסו בעקבותיו גם היישומים האחרים.
- ניתן לעצור יישום בודד (לפרוק מהזיכרון את כל מרכיביו) מבלי להשפיע על היישומים האחרים.

יישום ASP הוא לא אותו הדבר כמו אתר אינטרנט. אתר אינטרנט יכול להכיל מספר יישומים המאוחסנים במיגוון ספריות. בדרך כלל, אתה יוצר יישומים נפרדים כאשר יש לך אוספים של דפים המשויכים למשימות נפרדות. לדוגמה, ייתכן ותיצור יישום אחד המכיל את כל הדפים המיועדים לשימוש הקהל הרחב, ויישום נפרד המיועד לשימושם של מנהלי האתר בלבד.

יישום מוגדר על ידי השימוש ב-Internet Service Manager לציון ספריית שורש עבורו. יישום מורכב מספריה מסוימת וכל ספריות המשנה שלה. אם אחת מספריות המשנה הללו מוגדרת גם היא להיות יישום בפני עצמה, היא מרכיבה יישום נפרד. במילים אחרות, אין שני יישומים חופפים.

כאשר אתה מתקין דפי ASP בפעם הראשונה, נוצרים מספר יישומי ברירת מחדל. לדוגמה, נוצר יישום עבור אתר ברירת המחדל שלך. אבל, אתה יכול ליצור יישומים נוספים לפי צרכיך.

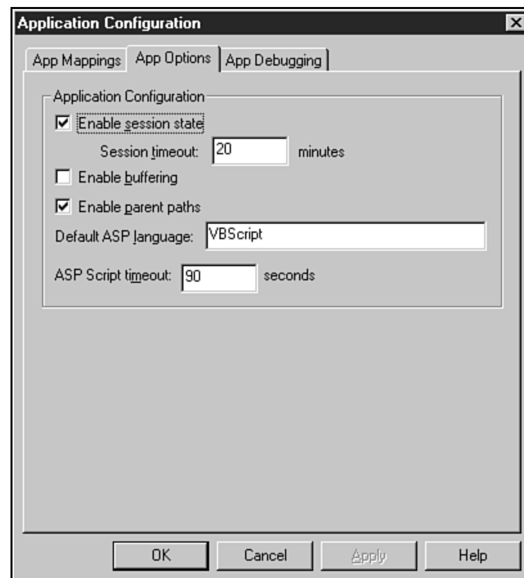
עקוב אחר הצעדים הבאים להגדרת יישום ASP:

1. הפעל את Internet Service Manager מתוך קבוצת היישומים Microsoft Internet Information Server.
2. לחץ על שם אתר ברירת המחדל שלך בעץ הניווט (אם לא שינית דבר בעת ההתקנה ייקרא אתר זה Default Web Site).
3. באפשרותך לבחור כל ספריה קיימת, לבחור באתר ברירת המחדל, או ליצור ספריה חדשה עבור היישום שלך. כדי ליצור ספריה וירטואלית חדשה, לחץ לחיצה ימנית על שם אתר ברירת המחדל ומתפריט הקיצור הצבע על Create New ובחר Virtual Directory.
4. לאחר שבחרת את הספריה עבור היישום שלך, עליך לסקור את מאפייניה. תוכל לעשות זאת על ידי לחיצה על לחצן Properties בסרגל הכלים, או על ידי לחיצה ימנית על שם התיקיה ומתפריט הקיצור לבחור Properties.
5. בתיבת דו-שיח מאפייני ספריה בחר בכרטיסיה Home Directory או Virtual Directory.
6. בחלק Application Settings, לחץ על Create.

**הערה:**

כאשר אתה יוצר יישום אתה יכול לקבוע עבורו שם. שם זה משמש רק בעת העבודה עם Internet Service Manager. הוא אינו משפיע על תסריטי ASP שלך.

יצרת בהצלחה יישום חדש. לאחר שעשית זאת, תוכל להגדיר חלק ממאפייניו על ידי בחירה באפשרות Configuration מ- Application Settings (ראה תרשים 4.1). לדוגמה, תוכל לציין האם היישום משתמש ב-Sessions או אם היישום אמור לפעול במרחב זיכרון משל עצמו.



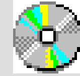

**תרשים 4.1** יצירת יישום

# יישומים והקובץ Global.asa

יישומי ASP יכולים להשתמש בקובץ מיוחד בשם Global.asa. כל יישום יכול לכלול רק קובץ אחד כזה, והוא חייב להיות מאוחסן בספריית השורש של היישום. קובץ זה מכיל מידע כוללני לכל היישום. קובץ Global.asa נראה כך:

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Application_OnStart
END SUB
</SCRIPT>

<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Application_OnEnd
END SUB
</SCRIPT>

<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Session_OnStart
END SUB
</SCRIPT>

<SCRIPT LANGUAGE=VBScript RUNAT=Server>
 SUB Session_OnEnd
 END SUB
</SCRIPT>
```

**הערה:**



ניתן למצוא קובץ Global.asa תקני בתקליטור המצורף לספר זה, בתיקיה \Books\59303\Chapter04\ChatPage.

קובץ Global.asa מכיל שתי שגרות המופעלות על ידי אירועי יישום. השגרה Application_OnStart מופעלת כאשר היישום מופעל לראשונה, ואילו השגרה Application_OnEnd מופעלת כאשר מסתיימת פעולת היישום. ניתן לשייך רק תסריט אחד לכל אחד משני האירועים.

השגרה Application_OnStart מופעלת פעם אחת כאשר מתקבלת בקשה ראשונה לדף הקיים ביישום זה. האירוע אינו מופעל פעם נוספת עד אחרי ששירות IIS נעצר, נערכים שינויים בקובץ Global.asa, או כאשר היישום מוסר מהזיכרון באמצעות Internet Service Manager.

שים לב שכדי לציין תסריט משתמש הקובץ Global.asa בתחביר המורחב של התגית <SCRIPT> של Microsoft. אתה חייב להשתמש בשיטה זו לציון תסריט בקובץ Global.asa, ולא להשתמש בתוחמים <% ו- %> הרגילים. הקובץ Global.asa שבדוגמה הקודמת עושה שימוש בשפת VBScript, אבל ניתן להשתמש בשפות תסריט שונות.

בקובץ Global.asa אין אפשרות לכלול פלט מכל סוג שהוא. במיוחד, אינך יכול להשתמש בתגיות HTML כלשהן או בשיטה ()Response.Write. הקובץ Global.asa עצמו אינו מוצג לעולם. הוא קיים רק כדי להכיל תסריטים ואובייקטים.

כדי ליצור תסריט המופעל כאשר היישום מופעל, עליך פשוט להוסיף את התסריט לשגרה Application_OnStart, כפי שמוצג בדוגמה הבאה:

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Application_OnStart
    Dim FileName, MyFS, MyFile
    Set MyFS = Server.CreateObject( "Scripting.FileSystemObject" )
    Set MyFile = MyFS.OpenTextFile( "c:\applog.txt", 8, TRUE )
    MyFile.WriteLine NOW() & " - Application Started"
    MyFile.Close
    Set MyFile = Nothing
    Set MyFS = Nothing
END SUB
</SCRIPT>
```

תסריט זה פשוט רושם לקובץ יומן (Log File) בשם applog.txt את התאריך והשעה בה הופעל היישום. תוכל להיעזר בקובץ יומן זה לשם מעקב אחר הזמנים בהם נדרש להפעיל את היישום שלך מחדש (למשל, לאחר שהשרת קרס).

**הערה:**

התסריט שבדוגמה זו משתמש ברכיב הגישה לקובץ (File Access Component) כדי לכתוב לקובץ טקסט. כדי ללמוד עוד אודות השימוש ברכיב זה קרא את פרק 8.

הקובץ Global.asa יכול גם לשמש להצהרת אובייקטים עם טווח יישום (כדי ללמוד עוד בנושא זה, קרא את הסעיף שכותרתו **שילוב אובייקטים ורכיבים בדפי ASP** בפרק 1). לדוגמה, תוכל להצהיר על מופע של האובייקט Dictionary, הזמין עבור כל הדפים בתוך היישום, על ידי השימוש בקוד HTML הבא:

```
<OBJECT RUNAT="Server" SCOPE="Application" ID="MyDictionary"
PROGID="Scripting.Dictionary"></OBJECT>
```

כאשר אתה כולל קוד HTML מעל קובץ Global.asa עליך לכלול אותו מחוץ לשגרות או תגיות תסריט כלשהן. לדוגמה, תוכל להציב את הצהרות האובייקט שלך בחלקו העליון של הקובץ Global.asa.

**אזהרה:**

כאשר אתה מצהיר על אובייקט עם טווח יישום, היזהר ממודל המטלות (Threading Model). הצהרה על מופע של אובייקט המשתמש במודל בניית מטלות על-פי דירות (Apartment-Threading Model), כמו למשל האובייקט Dictionary, יכול להשפיע על ביצועי אתר אינטרנט בו קיימת כמות גבוהה של תעבורה. וגרוע אף מכך, עלולות להיווצר הודעות שגיאה.

לאחר שאתה מצהיר על מופע של אובייקט בקובץ Global.asa באמצעות התגית <OBJECT> אתה יכול להשתמש באובייקט זה בכל אחד מדפי ASP שלך, ממש כאילו היה זה אובייקט מובנה של ASP. אינך צריך להצהיר על אובייקט זה פעם נוספת.

# האובייקט Application

לאובייקט Application יש את כל השיטות והאוספים המשויכים ליישומים. בסעיפים הבאים תלמד כיצד להשתמש באובייקט Application כדי ליצור משתני יישום, וכיצד להשתמש במשתנים אלה בקובץ Global.asa.

## שימוש במשתני יישום

משתנה יישום (Application Variable) מכיל נתונים בהם ניתן להשתמש בכל הדפים ועל ידי כל המשתמשים ביישום. משתני יישום יכולים להכיל כל סוג נתון, כולל מערכים (Arrays) ואובייקטים. הנה רשימה של שימושים שכיחים למשתני יישום:

- משתנה יישום (Application Variable) יכול לשמש להצגת מידע ארעי בכל דף אינטרנט. למשל, תוכל להשתמש במשתנה יישום להצגת "הטיפ היומי" או חדשות עדכניות בכל דף אינטרנט.
- משתנה יישום (Application Variable) יכול לרשום את מספר הפעמים בהן לחץ משתמש על כרזת פרסום (Banner) כלשהי באתר שלך (תלמד כיצד לעשות זאת בפרק 9).
- משתנה יישום (Application Variable) יכול לאחסן מידע שאוחזר מבסיס נתונים. למשל, אתה יכול לאחזר רשימת פריטים למכירה באתר שלך, ולהציג את הרשימה במספר דפים תוך שימוש במשתנה יישום (Application Variable).
- משתנה יישום (Application Variable) יכול להכיל מונה רץ של מספר המבקרים באתר שלך. תלמד כיצד לעשות זאת בסעיף, **הדף WhosOn**, שבהמשך הפרק.
- משתנה יישום (Application Variable) יכול לשמש לאפשור תקשורת בין משתמשים באתר שלך. למשל, תוכל להיעזר במשתני יישום כדי ליצור משחקים רבי-משתתפים או חדרי שיחה רבי משתמשים. דוגמה לדף Chat כזה, המשתמש במשתני יישום תוצג בסעיף הבא של פרק זה, **דף Chat**.

## יצירה וקריאה של משתני יישום

יצירת משתני יישום היא פעולה קלה. כדי ליצור משתנה יישום (Application Variable) חדש אתה יכול פשוט להעביר את שם המשתנה החדש לאובייקט Application, כמו בדוגמה הבאה:

```
<HTML>
<HEAD><TITLE> Application Example </TITLE></HEAD>
<BODY>
<%
Application("Greeting")="Welcome!"
%>
<%=Application("Greeting")%>
</BODY>
</HTML>
```

בדוגמה זו נוצר משתנה יישום (Application Variable) חדש בשם Greeting ומשויך לו הערך "Welcome!". אחר כך, מועבר ערכו לדף המוצג בדפדפן.

לאחר שלמשתנה יישום (Application Variable) שויך ערך, ניתן להציג את הערך בכל הדפים של היישום. לדוגמה, הדף הבא יציג גם הוא את הברכות, למרות שלמשתנה Greeting לא שויך ערך כלשהו בדף זה:

```
<HTML>
<HEAD><TITLE> Another Page </TITLE></HEAD>
<BODY>
<%=Application("Greeting")%>
</BODY>
</HTML>
```

לאחר ששויך ערך למשתנה יישום (Application Variable) הוא מחזיק בערך זה עד אשר שרת האינטרנט נסגר, או עד שהיישום מוסר מהזיכרון. אם אתה בר מזל, יכול הדבר להיארך שבועות, ואף חודשים. מכיון שמשתני יישום אינם מפורקים באופן אוטומטי כאשר המשתמש עוזב, עליך להיזהר ולא להשתולל בעת יצירתם. משתני יישום גוזלים משאבי זיכרון; עליך להשתמש בהם בצמצום.

### הערה:

איך אתה מסיר משתנה יישום (Application Variable) לאחר שהוא נוצר? בעת השימוש בגירסה 4 של IIS אינך יכול לעשות זאת. משתני יישום נשארים בזיכרון עד אשר השרת מכובה, או שמתבצעים שינויים בקובץ Global.asa, או עד שהיישום מוסר מהזיכרון. הגירסה החדשה של דפי ASP הנכללת במערכת ההפעלה Windows 2000 מכילה שיטות חדשות להסרת משתני יישום.

חשוב גם להבין שמשתנה יישום (Application Variable) אינו יחסי למשתמש מסוים. אם משתמש אחד מבקש דף אינטרנט המשייך ערך אחד למשתנה יישום, ומשתמש אחר מבקש דף המשייך ערך אחר, ערך המשתנה ישתנה עבור שני המשתמשים. ראה את התסריט הבא:

```
<%
Randomize
IF INT(2*RND)=1 THEN
    Application("FavoriteColor")="Blue"
ELSE
    Application("FavoriteColor")="Red"
END IF
Response.Write Application( "FavoriteColor" )
%>
```

תסריט זה משייך באופן אקראי את הערך Red או את הערך Blue למשתנה היישום (Application Variable) בשם FavoriteColor. נניח ששני משתמשים מאחזרים את הדף המכיל את התסריט הזה. במקרה שכזה, ערך המשתנה יהיה זהה עבור שני המשתמשים. למשתנה יהיה הערך ששויך לו כאשר המשתמש השני אחזר את הדף.

דבר זה עלול לגרום לבעיה. מכיון שיותר ממשתמש אחד יכול לגשת למשתנה יישום (Application Variable) בו-זמנית, עלולות להיווצר התנגשויות. למשל, נניח שאתה משתמש במשתנה יישום לרישום מספר הפעמים בהם לחצו מבקרים באתר על כרזת פרסום. בכל פעם שמשתמש לוחץ על הכרזה מופעל תסריט כגון זה :

```
<%
NumClicks=Application("BannerClicks")
NumClicks=NumClicks+1
Application("BannerClicks")=NumClicks
%>
```

תסריט זה פשוט מקדם באחד מספר המאוחסן במשתנה היישום (Application Variable) בשם BannerClicks. אבל, נניח ששני משתמשים לחצו על כרזת הפרסום בו-זמנית. אותו תסריט יופעל בו-זמנית עבור שני המשתמשים. אם מתרחש כדבר הזה, יהיה ערכו של המשתנה BannerClicks לא מדויק. שני המשתמשים קידמו את ערך המשתנה לאותו ערך.

למרבה המזל, לאובייקט Application יש שתי שיטות היכולות לסייע בפתרון מצב כגון זה בדיוק. השיטות Lock ו- Unlock משמשות למנוע באופן זמני מהמשתמשים לשנות את ערך משתנה היישום (Application Variable). להלן הדוגמה הקודמת, משוכתבת כך שתמנע התנגשות אפשרית :

```
<%
Application.Lock
NumClicks=Application("BannerClicks")
NumClicks=NumClicks+1
Application("BannerClicks")=NumClicks
Application.Unlock
%>
```

השורה הראשונה בתסריט זה נועלת את כל המשתנים באובייקט Application. כאשר המשתנים נעולים, משתמשים אחרים אינם יכולים לשנות אותם, עד שמוסרת הנעילה. משתני היישום נשארים נעולים עד אשר נקראת השיטה Unlock באופן מפורש (כמו בדוגמה זו), או עד שמגיע סוף הדף.

שים לב שאינך יכול לבחור איזה משתנה יישום (Application Variable) לנעול; הבחירה היא בין הכל או לא כלום. התסריט הקודם מונע באופן זמני משתמשים אחרים משינוי כל משתני היישום הקיימים.

חשוב להבין שנעילת משתני היישום אינה מונעת באופן קבוע מהמשתמשים לשנות את ערכי המשתנים. הנעילה פשוט מחייבת שכל שינוי יתבצע באופן מסודר ומאורגן. השינוי במשתני היישום מבוצע באופן טורי ומאורגן, ולא באי-סדר.

# הצגת משתני יישום

רוב משתני המערכת מאוחסנים בעצם באוסף Contents (Contents collection) של האובייקט Application. בכל פעם שאתה יוצר משתנה יישום (Application Variable) חדש, נוסף לאוסף (collection) זה פריט חדש. למשל, שתי ההצהרות הבאות הן זהות :

```
<% Application("FavoriteColor")="Blue" %>
<% Application.Contents("FavoriteColor")="Blue" %>
```

מאחר ומשתני יישום מאוחסנים באוסף (Collection), אתה יכול לטפל בהם על ידי שימוש בכל מאפייני האוסף והשיטות בהם דנו קודם לכן. אתה יכול לאחזר מונה של מספר משתני היישום, תוך שימוש במאפיין Count. אתה יכול להציג את כל הפריטים המוכלים באוסף Contents, בין אם על ידי לולאת FOR...EACH ובין אם על ידי לולאת FOR...NEXT. הנה דוגמה לשימוש בלולאה FOR...EACH :

```
<%
FOR EACH thing IN Application.Contents
Response.Write("<BR>"&thing&"="&Application.Contents(thing))
NEXT
%>
```

תסריט זה יציג את כל משתני היישום, על ידי קריאה בלולאה של האוסף Application.Contents.

**הערה:**

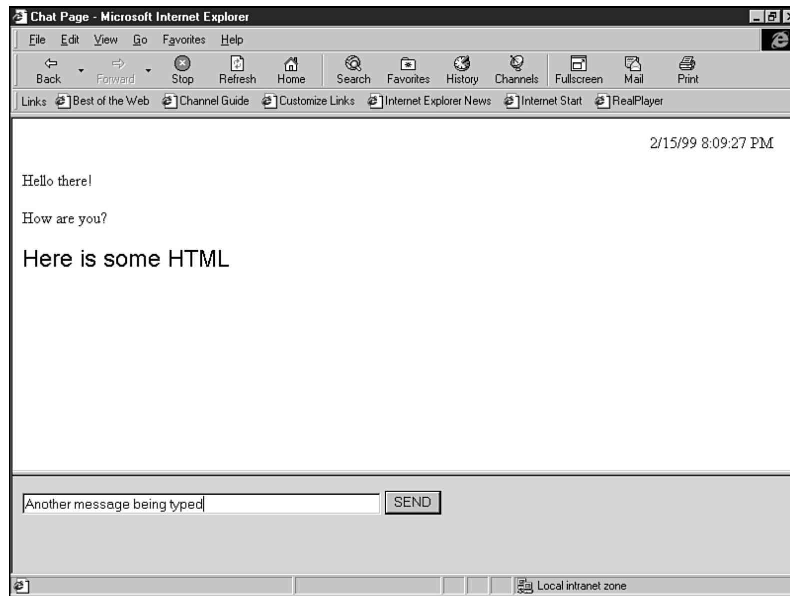
האובייקט Application כולל גם אוסף נוסף בשם StaticObjects. אוסף זה מכיל מידע אודות כל האובייקטים המוצהרים עם טווח יישום בקובץ Global.asa בעת השימוש בתגית <OBJECT> של HTML.

# דף Chat

בסעיף זה, ובבא אחריו, תלמד כיצד ליצור שני פרויקטים של ASP תוך שימוש באובייקט Application - דף Chat ותוכנית לסטטיסטיקה בזמן אמת של אתר אינטרנט. פרויקטים אלה נועדו לתאר חלק מאותם הנושאים הנידונים בפרק זה, אך אני מקווה שתמצא אותם מועילים גם בקשר לאתר האינטרנט שלך.

הדרך הטובה ביותר למשוך משתמשים שישובו לאתר שלך פעם אחר פעם היא ליצור תחושה של קהילה. אחת מהדרכים הטובות ביותר לעשות זאת היא למצוא את הדרך בה יכולים המבקרים באתר לתקשר אחד עם השני. דף Chat יעשה בדיוק את הדבר הנכון.

דף Chat הנידון בחלק זה מאפשר שיחה בזמן-אמת בין מספר משתתפים. כל המשתמשים אשר מבקשים דף זה יכולים לצפות בהודעות שהושמו בו על ידי משתמשים אחרים. מעבר לכך, הם יכולים גם להוסיף את ההודעות שלהם (ראה תרשים 4.2).




**תרשים 4.2** דף Chat

כדי ליצור דף Chat עליך ליצור שלושה קבצים. הרשימה הבאה היא סקירה של הקבצים אותם צריך ליצור או לשנות לצורך פרויקט זה:

- דף Chat - דף זה יכלול שתי מסגרות (Frames); העליונה תציג את כל ההודעות ממשתמשים אחרים והתחתונה תאפשר כתיבת הודעות חדשות.
- דף התצוגה - דף זה יציג את כל ההודעות המוקלדות על ידי משתמשים אחרים. כל חמש שניות לערך יעודכן תוכן מסגרת זו כדי להציג הודעות חדשות.
- דף ההודעות - דף זה יאפשר למשתמש להקליד הודעה חדשה. הוא מכיל תיבת טקסט יחידה.
- קובץ Global.asa - בקובץ זה יתוקן התסריט Application_OnStart.

# יצירת דף Chat

הדף הראשון שתיצור הוא דף Chat. המטרה היחידה של דף זה היא לפעול כמכולה (Container) עבור שני הדפים האחרים. מכיון שדף זה אינו כולל תסריט כלשהו, עליך לשמור אותו בשם ChatPage.asp, כפי שמוצג בתדפיס 4.1.

**תדפיס 4.1** קוד הדף ChatPage.asp

```
<HTML>
<HEAD><TITLE> Chat Page </TITLE></HEAD>
<FRAMESET ROWS="*,100">
<FRAME SRC="display.asp">
<FRAME SRC="message.asp">
</FRAMESET >
</HTML>
```

מדוע בכלל עלינו ליצור מסגרות? מאחר ודף התצוגה מתרענן באופן אוטומטי כל חמש שניות, נדרש דף נפרד להקלדת הודעה חדשה. אחרת, עלול הדף להתרענן כאשר המשתמש נמצא באמצע כתיבת ההודעה שלו (מה שיכול להיות מרגיז למדי).

**טיפ:**

אם אתה מתכוון להציב את אתר האינטרנט שלך במחשב מבוסס UNIX, רצוי מאוד שתקפיד שכל שמות הקבצים וכל התייחסות לשמות הקבצים (בקישורים ביניהם) יופיעו באותיות קטנות (Lowercase). UNIX היא מערכת הפעלה הרגישה לאותיות רישיות (Case Sensitive), בעוד שמערכות Windows אינן כאלו (Case InSensitive).

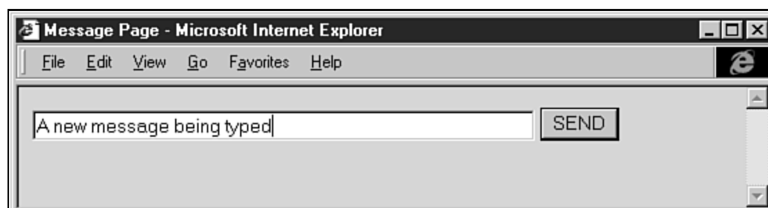
# תיקון הקובץ Global.asa לצרכי דף Chat

כדי שדף Chat יעבוד, צריך לבצע שינויים בקובץ Global.asa. התסריט הבא משמש לאתחול משתני היישום הדרושים עבור דף Chat. המשתנים חייבים להיות משתני יישום, כדי שיוכלו להיות נגישים עבור כל המשתמשים. משתנה היישום (Application Variable) הראשון שמו Talk. זהו מערך (Array) שמחזיק את כל ההודעות. המערך Talk נוצר על ידי כך שמשייכים אליו את TempArray. משתנה היישום השני נקרא Tplace (קיצור של TalkPlace) והוא משמש להצביע על ההודעה הנוכחית במערך Talk. התסריט הבא מאפס את המשתנה הזה:

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Application_OnStart
    Dim TempArray(5)
    Application("Talk")=TempArray
    Application("Tplace")=0
END SUB
</SCRIPT>
```

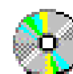
# יצירת דף ההודעות

מטרת דף ההודעות היא לאפשר למשתמש להקליד הודעה חדשה (ראה תרשים 4.3). הדף מכיל טופס HTML ובו תיבת קלט טקסט בודדה, ולחצן הגשה (Submit). כאשר נלחץ הלחצן SEND הדף טוען את עצמו מחדש.




**תרשים 4.3** דף ההודעות

תסריט זה מבצע שני דברים. ראשית, הוא בודק האם קיימות יותר מאשר ארבע הודעות. אם יש יותר מארבע הודעות מאופס משתנה היישום (Application Variable) TPlace לאפס. דבר זה מונע מהמערך Talk מלזלוג עם יותר מדי הודעות.



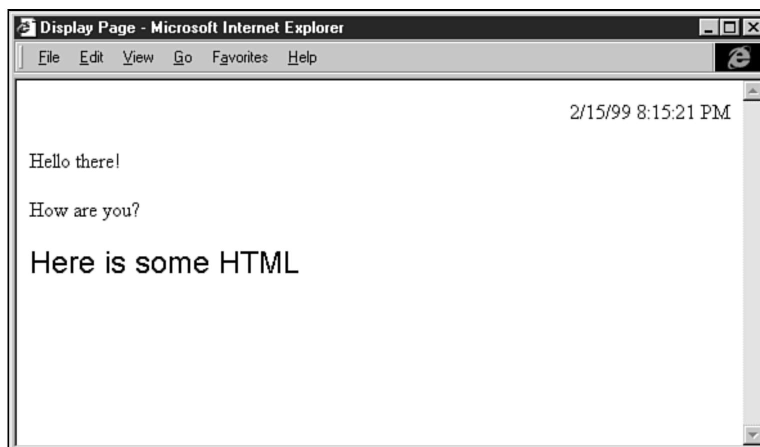
אחר כך, מוסיף התסריט את ההודעה החדשה למערך Talk ומקדם את הערך של TPlace. TPlace לעולם יצביע על המיקום הבא במערך Talk, המקום בו ניתן להכניס הודעה חדשה. הקוד המלא של דף ההודעות מופיע בתדפיס 4.2 (תמצא את הקוד בתקליטור המצורף לספר, קובץ בשם message.asp):

**תדפיס 4.2** תדפיס עבור MESSAGE.ASP

```
<%
IF not Request.Form("message")="" THEN
  Application.LOCK
  IF Application("TPlace")>4 THEN
     Application("TPlace")=0
  END IF
   TempArray=Application("Talk")
   TempArray(Application("TPLACE"))=Request.Form("Message")
   Application("Talk")=TempArray
   Application("TPlace")=Application("TPlace")+1
  Application.Unlock
END IF
%>
<HTML>
<HEAD><TITLE> Message Page </TITLE></HEAD>
<BODY BGCOLOR="LIGHTBLUE">
<FORM METHOD="POST" ACTION="message.asp">
<INPUT NAME="message" TYPE="TEXT" SIZE=50>
<INPUT TYPE="SUBMIT" VALUE="SEND">
</FORM>
</BODY>
</HTML>
```

# יצירת דף התצוגה

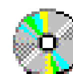
הדף האחרון שיש ליצור הוא דף התצוגה. בדף זה מוצגות בפועל כל הודעות המשתמשים (ראה תרשים 4.4).




**תרשים 4.4** דף התצוגה

הדף מרענן את עצמו באופן אוטומטי כל חמש שניות. הוא עושה זאת תוך שימוש ב-Client pull (משיכת לקוח). תגית <META> של HTML כוללת פקודה המבצעת זאת (היא מוסיפה את הכותרת REFRESH לדף ASP).

התסריט הראשון בדף זה משמש לזיהוי הדף הנוכחי. כתובת URL המלאה של הדף מאוחזרת מהאוסף ServerVariable, ומשויכת למשתנה ששמו MySelf. המשתנה MySelf משמש בתוך התגית <META> כדי לציין שהדף יתרענן.



התסריט העיקרי משמש להצגת תוכן המערך Talk. הלולאה FOR...NEXT מציגה את כל ההודעות הנוכחיות. הקוד של דף התצוגה מוצג בתדפיס 4.3 (תמצא את הקוד בתקליטור המצורף לספר, קובץ בשם display.asp):

**תדפיס 4.3** קוד דף התצוגה

```
<%
MyServer=Request.ServerVariables("SERVER_NAME")
MyPath=Request.ServerVariables("SCRIPT_NAME")
MySelf="HTTP://"&MyServer&MyPath
%>
<HTML>
<HEAD>
<META HTTP-EQUIV="REFRESH" CONTENT="5;<%=MySelf%>">
<TITLE>Display Page</TITLE>
</HEAD>
<BODY>
<P ALIGN=RIGHT><%=NOW%></P>
<%
TempArray=Application("Talk")
FOR i=0 to Application("TPlace")-1
  Response.Write("<P>"&Temparray(i))
NEXT
%>
</BODY>
</HTML>
```

## הרחבת פרויקט דף Chat

קיימות מספר דרכים לשיפור דף Chat. למשל, המספר המירבי של הודעות אותן יכול הדף להציג בכל רגע נתון הוא חמש. במקרים של עומס הדבר לא יעיל. תוכל לשנות את מספר ההודעות המירבי, על ידי עריכת גודלו של המערך TempArray בקובץ Global.asa, ועל ידי שינוי המספר מאפס TPlace את דף ההודעות.

דף Chat גם מאפשר לך להוסיף הודעות הכוללות עיצובי HTML. אבל, ניתן להקל על ביצוע הדבר בפשטות. לדוגמה, תוכל להוסיף תיבות סימון לעיצוב ההודעה, אשר מאפשרות לך לציין גופן וצבע עבור ההודעה.

לסיכום, דף Chat אינו משייך שמות משתמשים עם ההודעות שהם שולחים אליו. שוב, זה לא קשה לשנות זאת. תוכל להוסיף גם דף כניסה (Logon) לפני דף Chat, אשר מבקש שם משתמש. שם זה ניתן אז לצרף כקידומת לכל ההודעות שמשתמש זה שולח.

**הערה:**

תוכל לראות גרסאות עובדות של רוב הדוגמאות בספר זה על ידי ביקור באתר המלווה את הספר, בכתובת http://www.aspsite.com.

# דף WhosOn

למנהלי אתרים טובים יש מטרה עיקרית אחת בחיים: הם מעוניינים ביותר מבקרים באתרי האינטרנט שלהם. חלק מהאובססיה הזו כולל גם מעקב מדויק אחר מספר המשתמשים המחוברים כרגע לאתר, ומה הם עושים בו.

הפרויקט המתואר בסעיף זה של הפרק מאפשר לך לעקוב אחר המבקרים באתר האינטרנט שלך בזמן-אמת. אתה יכול לדעת את מספר המבקרים באתר שלך בכל רגע נתון. אתה גם יכול לדעת איזה דף פתח כל אחד מהמבקרים (ראה תרשים 4.5).

WhosOn

| User ID | Current Page | IP Address | Start Time |
|---|---|---|---|
| 373038728 | /whoson.asp | 127.0.0.1 | 12/2/99 12:27:59 AM |
| 373038734 | /chat.asp | 187.123.87.1 | 12/2/99 6:16:01 PM |
| 373038722 | /code/sample23.asp | 178.92.54.1 | 12/2/99 6:14:28 PM |
| 373038719 | /forums/talk.asp | 176.0.8.43 | 12/2/99 6:22:53 PM |
| 373038706 | /whoson.asp | 512.19.34.1 | 12/2/99 6:04:19 PM |
| 373038721 | /components/ad.asp | 255.132.0.1 | 12/2/99 5:09:12 AM |

**תרשים 4.5** הדף WhosOn

הפרויקט מדגים כיצד לשייך מערך יישום (Application Array) למשתנה יישום (Application Variable). מערך יישום משמש לאחסון מידע אודות המבקרים. כאשר משתמש מבקש דף כלשהו, יעודכן המידע במערך היישום. יש ליצור או לעדכן את הקבצים הבאים עבור פרויקט זה:

- הקובץ Global.asa - שני התסריטים, Application_OnStart ו- Application_OnEnd, חייבים להתעדכן לצורך פרויקט זה.
- הקובץ GrabStats - קובץ זה מעדכן את מערך היישום. יהיה עליך לכלול קובץ זה בכל דף אחרי אתה מעוניין לעקוב.
- הדף WhosOn - דף זה מציג את המשתמשים המחוברים כרגע לאתר האינטרנט שלך.

## הערה:

במהדורה הראשונה של ספר זה נעשה שימוש באובייקט Dictionary לאחסון המידע אודות משתמשים המחוברים לאתר ואשר מנוטרים על ידי הדף WhosOn. בשלב מסוים דירבנה חברת Microsoft מפתחים להשתמש באובייקט Dictionary עם טווח יישום כדי להטמין מידע ולשפר את יעילות אתרי האינטרנט (חברת Microsoft עצמה השתמשה באובייקט Dictionary באופן זה כדי לשפר את ביצועי אתר האינטרנט שלה עצמה, www.microsoft.com).

אולם, המשתמשים החלו נתקלים בשגיאות בעת השימוש באובייקט Dictionary עם טווח יישום (במקרים של תנועת משתמשים כבדה הופיעה ההודעה A trappable

error has occurred in an external object בכל רחבי האתר). מסתבר שהאובייקט Dictionary מעולם לא יועד לשימוש בטווחי session או יישום בשרת. וכדי להוסיף לבלבול, האובייקט Dictionary סומן על ידי מודל מטלות (Threading Model) שגוי ברישום המערכת (Registry).

בעודי כותב שורות אלו, שחררה חברת Microsoft רכיב חדש בשם LookupTable המחליף את מרבית הפעילויות של האובייקט Dictionary. רכיב זה נועד במיוחד לשימוש בטווחי session או יישום.

למרות זאת, במהדורה זו של הספר, נכתב הדף WhosOn מחדש תוך שימוש במערך יישום, ולא עם הרכיב LookupTable, מכיון שהרכיב LookupTable הוא לקריאה בלבד (Read Only). כדי להוריד את הרכיב ו/או לקרוא לגביו פרטים נוספים, פנה לאתר האינטרנט http://msdn.microsoft.com.

# תיקון הקובץ Global.asa לצרכי הדף WhosOn

כדי ליצור פרויקט זה עליך לשנות שני תסריטים בקובץ Global.asa. ראשית, עליך ליצור מערך יישום, המשמש לאחסון מידע אודות המבקרים באתר האינטרנט שלך. מכיון שמערך זה חייב להיווצר פעם אחת בלבד, מתבצע הדבר בתסריט Application_OnStart:

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Application_OnStart
    DIM Stats( 100, 3 )
    Application ( "Stats" ) = Stats
END SUB
</SCRIPT>
```

לתסריט הקודם הוספו שתי שורות. ההצהרה הראשונה מצהירה על מערך דו-מימדי בשם Stats. ההצהרה השנייה משייכת את המערך למשתנה יישום (Application Variable) בשם זהה. לאחר שהוא נוצר יכול משתנה זה לשמש בכל מקום ביישום.

**הערה:**

בפרק 5 תלמד כל מה שצריך אודות האובייקט Session.

התסריט Application_OnEnd שבקובץ Global.asa חייב גם הוא להיות מתוקן. מטרת התסריט הבא היא להסיר משתמש מהמערך Stats כאשר Session המשתמש מסתיים (כברירת מחדל קורה הדבר לאחר 20 דקות של חוסר פעילות מצידו):

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Application_OnEnd
    Application.Lock
    localStats = Application( "Stats" )
    FOR i = 0 TO UBOUND( localStats, 2 )
        IF localStats( 0, i ) = Session.SessionID THEN
            localStats( 0, i ) =  " "
            EXIT FOR
        END IF
    NEXT
    Application ( "Stats" ) = localStats
    Application.Unlock
END SUB
</SCRIPT>
```

המעקב אחר משתמשים יתבצע באמצעות מזהה ה-Session (Session ID). המימד הראשון במערך Stats משמש לאחסון מזהי ה-Session. התסריט הקודם בודק האם רשומה עבור מזהה ה-Session של המשתמש הנוכחי כבר קיימת במערך Stats. אם היא קיימת, היא מוסרת.

## יצירת הקובץ GrabStats

כדי לקבוע באיזה דף צופה כרגע המבקר באתר עליך לכלול קובץ בכל דף אחריו אתה מעוניין לעקוב. הקובץ כולל את התסריט שבתדפיס 4.4:

**תדפיס 4.4** קוד שיש להוסיף לקובץ GrabStats.asp

```
<%
foundUser = FALSE
Application.Lock
localStats = Application( "Stats" )
FOR k = 0 TO UBOUND( localStats, 2 )
    IF localStats( 0, k ) = Session.SessionID THEN
        localStats( 1, k ) =  Request.ServerVariables( "SCRIPT_NAME" )
        foundUser = TRUE
        EXIT FOR
    END IF
```

```
NEXT
IF foundUser = FALSE THEN
FOR k = 0 TO UBOUND( localStats, 2 )
    IF localStats( 0, k ) = "" THEN
        localStats( 0, k ) = Session.SessionID
        localStats( 1, k ) = Request.ServerVariables( "SCRIPT_NAME" )
        localStats( 2, k ) = Request.ServerVariables( "REMOTE_ADDR" )
        localStats( 3, k ) = NOW()
        EXIT FOR
    END IF
NEXT
END IF
Application("Stats") = localStats
Application.UnLock
%>
```

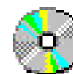
תסריט זה בודק האם מזהה Session עבור המשתמש הנוכחי כבר קיים במערך היישום Stats. אם המערך כבר כולל מזהה עבור Session זה, מעודכן המידע לגבי משתמש זה. אחרת, אם זהו הדף הראשון שביקש המשתמש מאתר האינטרנט, נוספת למערך רשומה חדשה.

הפריטים SCRIPT_NAME ו-REMOTE_ADDR מהאוסף ServerVariables משמשים כדי לאחזר את שם התסריט הנוכחי ואת כתובת IP של המשתמש. הפונקציה NOW() של VBScript משמשת להוספת תאריך ושעה לרשומה החדשה שנוספה.

שמור את הקובץ בשם GrabStats.asp. עליך לכלול (INCLUDE) קובץ זה בכל דף ASP אחריו אתה מעוניין לעקוב. הוסף את השורה הבאה בחלקו העליון של כל דף ASP:

```
<!-- #INCLUDE FILE="GrabStats.asp" -->
```

## יצירת הדף WhosOn



הדף WhosOn משמש להצגת מבקרים נוכחיים באתר האינטרנט. לגבי כל משתמש המבקר באתר מוצגים הפרטים הבאים: מזהה ה-Session (Session ID), הדף האחרון שהתבקש, כתובת IP ושעת תחילת הביקור. תדפיס 4.5 הוא הקוד עבור הדף WhosOn (תמצא את הקוד בתקליטור המצורף לספר, קובץ בשם WhosOn.asp):

**תדפיס 4.5** הקוד עבור הדף WhosOn

```
<!-- #INCLUDE FILE="GrabStats.asp" -->
<%
MyServer=Request.ServerVariables("SERVER_NAME")
MyPath=Request.ServerVariables("SCRIPT_NAME")
MySelf="HTTP://"&MyServer&MyPath
%>
<HTML>
<HEAD>
<META HTTP-EQUIV="REFRESH" CONTENT="20;<%=MySelf%>">
<TITLE>WHOSON</TITLE>
</HEAD>
```

```
<BODY>
<%
Application.Lock
localStats=Application("Stats")
Application.UnLock
%>
<CENTER>
<h2>WhosOn</h2>
<TABLE BORDER=1 CELLPADDING=10
CELLSPACING=0 BGCOLOR="#eeeeee">
<TR BGCOLOR="#cccccc">
    <TH>User ID</TH>
    <TH>Current Page</TH>
    <TH>IP Address</TH>
    <TH>Start Time</TH>
</TR>
<%
FOR i = 0 TO UBOUND( localStats, 2 )
IF localStats( 0, i ) <> "" THEN
%>
<TR>
    <TD><%=localStats( 0, i )%></TD>
    <TD><%=localStats( 1, i )%></TD>
    <TD><%=localStats( 2, i )%></TD>
    <TD><%=localStats( 3, i )%></TD>
</TR>
<%
END IF
NEXT
%>
</TABLE>
</CENTER>
</BODY>
</HTML>
```

השורה הראשונה בקובץ זה מכלילה את הקובץ GrabStats.asp שיצרת בסעיף הקודם. דבר זה מאפשר לך לדעת, כאשר אתה צופה בדף זה, שאתה אכן צופה בדף זה. הנתיב לדף WhosOn יופיע ליד מזהה ה-Session שלך.

התסריט הראשון משמש לאחזור נתיבו של הדף הנוכחי. הדף WhosOn מתרענן אוטומטית, תוך שימוש ב-Client Pull, מדי 20 שניות. נתיב הקובץ נדרש כדי שהדף יוכל להתרענן.

התסריט השני מעביר את המערך Stats המאוחסן במשתנה יישום (Application Variable) למערך זמני מקומי בשם localStats. מערך זה נהרס באופן אוטומטי כאשר הדף מסתיים.

התסריט האחרון מבצע לולאה על כל רשומות המערך localStats. אם לרשומה יש ערך, היא תוצג (כפי שניתן היה לראות בתרשים 4.5).

## הרחבת הדף WhosOn

כפי שהדברים כעת, הדף WhosOn מספק מידע חשוב אודות משתמשים באתר האינטרנט שלך. על ידי צפייה בדף WhosOn תוכל לבצע הערכה גסה של מספר המשתמשים המחוברים כרגע לאתר האינטרנט שלך. תוכל גם לראות את הדף האחרון בו צפה המבקר באתר.

קיימות מספר דרכים בהן ניתן להרחיב פרויקט זה. למשל, אם האתר שלך דורש ממשתמשים להירשם אליו, תוכל להציג את שם המשתמש, במקום את מזהה ה-Session שלו. כדי לעשות זאת, פשוט הוסף למערך Stats מרכיב חדש, עבור שם משתמש.

הדף WhosOn מוגבל להצגת מידע אודות מקסימום 100 משתמשים בכל רגע נתון. אם אתר האינטרנט שלך בורך בתנועה כבדה יותר, תוכל להגדיל את המערך בקובץ Global.asa. למשל, כדי לעקוב אחר 1,000 משתמשים, פשוט הצהר על המשתנה Stats באירוע Application_OnStart באופן הבא:

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Application_OnStart
    DIM Stats( 1000, 3 )
    Application ( "Stats" ) = Stats
END SUB
</SCRIPT>
```

**הערה:**

תוכל לצפות בגרסאות עובדות של חלק מהדוגמאות בספר זה על ידי ביקור באתר המלווה ספר זה, בכתובת http://www.aspsite.com.

# סיכום

בפרק זה למדת כיצד לעבוד עם יישומי ASP. החלק הראשון סיפק סקירה כללית של יישומים (Applications). בחלקו השני של הפרק למדת כיצד להשתמש באוספים ובמאפיינים של האובייקט Application. לסיום, בחלקו השלישי של הפרק, הוצגו בפניך שתי דוגמאות תכנות. למדת כיצד ליצור דף Chat ודף המאפשר לך לעקוב אחר המבקרים באתר האינטרנט שלך.

## פרק 5

# עבודה עם Sessions

**בפרק זה:**

- **מבוא ל-Sessions**
- **תפארת ה-Session**
- **עוגיות (Cookies, אבל לא מהסוג הטעים)**
- **שמירת מצב, ללא עוגיות (Cookies)**

בפרק זה תלמד כיצד לעבוד עם Sessions. תלמד כיצד להשתמש באוספים (Collections), מאפיינים (Properties), שיטות (Methods) ואירועים (Events) של session. בנוסף תלמד כיצד לקרוא וליצור Cookies. לסיום, יידונו מספר שיטות לשמירת מצב, גם ללא Cookies.

# מבוא ל-sessions

Session מתחיל ברגע שמשתמש מבקש דף כלשהו מאתר האינטרנט שלך ומסתיים מעט אחרי שמשתמש זה עוזב את האתר. כל מבקר באתר שלך מקבל session ייחודי.

Sessions יכולים לשמש לאחסון העדפות המשתמש. למשל, האם המשתמש מעדיף שלדפי האינטרנט בהם הוא צופה יהיה רקע ירוק, או רקע כחול? האם המשתמש מתנגד להצגת מסגרות (Frames)? האם המשתמש מעדיף לצפות בגירסה הטקסטואלית של האתר, במקום בגירסה הגרפית שלו? ניתן לעקוב אחר העדפות אלו ואחרות באמצעות session.

sessions יכולים גם לשמש ליצירת עגלת קניות וירטואלית. בכל פעם שהמשתמש בוחר בפריט לקנייה באתר האינטרנט שלך, ניתן להוסיף את המוצר לעגלת הקניות שלו. כאשר המשתמש מוכן לעזיבת האתר הוא יכול לרכוש את כל תוכן עגלת הקניות בקנייה אחת. כל הפרטים אודות הפריטים או המוצרים שבעגלת הקניות יכולים להישמר ב-session.

לבסוף, sessions יכולים לשמש למעקב אחר הרגליהם של המשתמשים באתר שלך. באותו אופן בו פעילי איכות הסביבה נעזרים בציוד מעקב כדי למפות את הרגלי הנדידה של הכריש הלבן, תוכל אתה להשתמש ב-sessions כדי לעקוב אחר תנועותיהם של המבקרים באתר בעודם נודדים בין הדפים בו. המידע יכול לשמש למטרות פרסום, לשיפור עיצוב האתר שלך, או פשוט לספק את סקרנותך.

# תפארת ה-session

ה-sessions נוצרו כדי למלא את מקומה של מגבלה הקיימת בפרוטוקול HTTP. זכור כיצד פועל פרוטוקול HTTP - בכל פעם שהמשתמש מבצע בקשה, השרת מספק תגובה. כל התקשורת בין הדפדפן ושרת האינטרנט מתבצעת בצמדי בקשה/תגובה לא רציפים.

שום חלק מהפרוטוקול HTTP אינו מאפשר לשרת לעקוב אחר משתמשים המבצעים בקשות. לאחר שהשרת סיים להגיב לבקשה, הוא אינו יכול להמשיך ולזהות את הדפדפן שביצע אותה. מנקודת מבטו של שרת האינטרנט, כל בקשה חדשה מבוצעת על ידי משתמש חדש. מסיבה זו נחשב פרוטוקול HTTP לפרוטוקול חסר-מצב (Stateless Protocol). פרוטוקול זה אינו יכול לשמש למעקב אחר מצבו של משתמש. זוהי מגבלה חמורה מפני שהיא גורמת לכך שאינך יכול לזהות משתמש המבקש מספר דפים.

לתיקון בעיה זו הוצגו Sessions. שימוש ב-sessions מאפשר לך לאחסן מידע אודות משתמש הצופה במספר דפים. sessions מאפשרים לך לבצע דברים רבים שבאופן אחר יהיו קשים מאוד לביצוע, או כלל בלתי אפשריים.

**אזהרה:**

שים לב ש- ASP sessions משתמשים ב- Cookies, ולא כל הדפדפנים תומכים בכך. לפרטים נוספים קרא בהמשך את הסעיף **כיצד באמת פועלים sessions**.

# אחסון מידע של session

משתני session (Session Variables) דומים למשתני יישום, בהם דן פרק 4. כמו משתני היישום (Application Variables), משתנה session יכול לשמש במספר דפי ASP. אולם, שלא כמו משתני יישום, לכל מבקר באתר נוצרים עותקים נפרדים משלו.

ASP Sessions קלים ביותר לשימוש. אתה יכול לשלוט בכל היבטי ה-session על ידי השימוש באובייקט Session של ASP. אם אתה צריך לאחסן נתונים שיהיו עקביים לכל אורך ה-session של המשתמש, עליך לאחסן את הנתונים לאוסף (collection) של האובייקט Session. להלן דוגמה:

```
<HTML>
<HEAD><TITLE> Session Example </TITLE></HEAD>
<BODY>
<%
Session("Greeting")="Welcome!"
Response.Write(Session("Greeting"))
%>
</BODY>
</HTML>
```

כאשר דף ASP זה מוצג בדפדפן מוצגת הברכה !Welcome. השורה הראשונה בתסריט משייכת את הטקסט !Welcome למשתנה session בשם Greeting. השורה השנייה בתסריט מציגה את הברכה על המסך.

עד כאן הדבר לא מלהיב במיוחד. הרי אתה יכול לעשות בדיוק את אותו הדבר גם עם משתנה VBScript פשוט. אבל, תאר לעצמך שאותו משתמש מבקש דף אחר, נוסף. לדוגמה, תאר לעצמך שהמשתמש מבקש את הדף הבא:

```
<HTML>
<HEAD><TITLE> Another Page </TITLE></HEAD>
<BODY>
<%=Session("Greeting")%>
</BODY>
</HTML>
```

כאשר המשתמש צופה בדף זה מוצגת הברכה !Welcome פעם נוספת. אבל, בדף זה לא שויך למשתנה ה-session (Session Variable) ערך כלשהו. Greeting, משתנה ה-session, שומר את הערך ששויך למשתנה בדף הקודם.

דבר זה אינו ניתן לביצוע באמצעות משתנה תסריט (Script Variable) רגיל. אורך חייו של משתנה רגיל הוא כאורך חייו של דף בודד. משתנה session, לעומת זאת, נשמר עד אשר המשתמש עוזב את האתר.

חשוב להבין שמשתני session קיימים רק באופן יחסי למשתמש מסוים. הערכים המשויכים למשתני session של משתמש אחד אינם משפיעים על ערכי משתני session של משתמש אחר. במילים אחרות, שלא כמו משתנה יישום (Application Variable), המידע המאוחסן במשתני session (Session Variable) אינו משותף בין המשתמשים השונים. לדוגמה, נניח שהתסריט הבא מופיע בדף ASP:

```
<%
Randomize
IF INT(2*RND)=1 THEN
    Session("FavoriteColor")="Blue"
ELSE
    Session("FavoriteColor")="Red"
END IF
%>
```

תסריט זה משייך באופן אקראי את הערך Blue או את הערך Red למשתנה FavoriteColor. למשתנה זה יכול להיות ערך שונה עבור משתמשים שונים. ערך המשתנה FavoriteColor אם כן הוא יחסי ל-session של משתמש מסוים.

מכיון שהערך של משתנה session אינו משותף עם מספר משתמשים, אין צורך לנעול את משתני ה-session, בדומה למשתני יישום. האובייקט Session אינו כולל את השיטות Lock או Unlock. לא מתרחשות התנגשויות, מפני שאין שני משתמשים שיש להם גישה לאותו משתנה session.

# תוכן ה-session

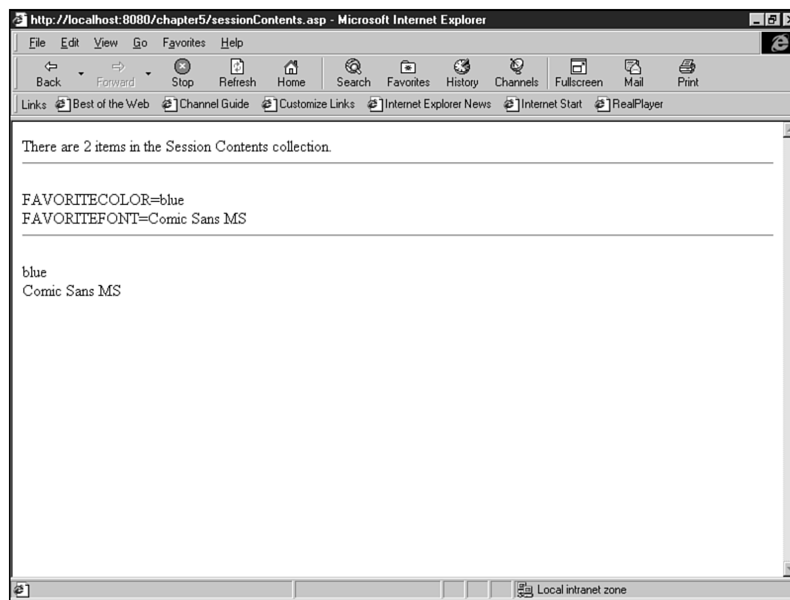
רוב משתני ה-session בעצם מאוחסנים באוסף של האובייקט Session ששמו Contents. לדוגמה, שתי ההצהרות הבאות הן זהות:

```
<%Session("MyVar")="Some Data" %>
<&Session.Content("MyVar") ="Some Data" %>
```

בדומה לאוספים בהם דנו קודם לכן, תוכל להשתמש במאפיין Count כדי לקבוע את מספר הפריטים באוסף Content. תוכל גם להציג את כל הפריטים שבאוסף Content באמצעות הלולאה FOR...EACH או הלולאה FOR...NEXT. דוגמה זו משתמשת בשתי השיטות (תמצא את הקוד בתקליטור המצורף לספר, קובץ בשם sessionContents.asp):

```
<%
Session("FavoriteColor")="blue"
Session("FavoriteFont")="Comic Sans MS"
%>
There are
<%=Session.Contents.Count %>
items in the Session Contents collection.
<HR>
<%
FOR EACH thing IN Session.Contents
  Response.Write( "<BR>"&thing&"="&Session.Contents(thing))
NEXT
Response.Write("<HR>")
FOR i=1 TO Session.Contents.Count
  Response.Write("<BR>"&Session.Contents(i))
NEXT
%>
```

בתסריט זה נוצרים שני משתני session, FavoriteColor ו- FavoriteFont. אחר כך, מאוחזר מספר הפריטים באוסף Contents. בסיום, מוצגים כל הפריטים שבאוסף באמצעות הלולאה FOR...EACH והלולאה FOR...NEXT. (ראה תרשים 5.1).



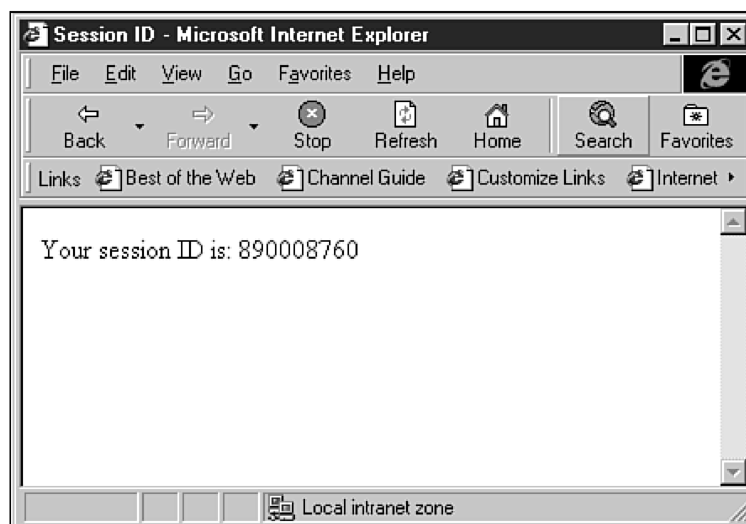

**תרשים 5.1** תוכן האוסף Contents

## זיהוי session

דפי ASP משייכים לכל session משתמש מזהה ייחודי (Unique Identifier). מזהה ה-session (Session ID) זה נוצר לראשונה כאשר נוצר session המשתמש, ומתמיד לכל אורכו של ה-session, כל עוד המשתמש נשאר באתר שלך. כדי לאחזר מזהה session עליך להשתמש במאפיין SessionID של האובייקט Session, כפי שתראה בדוגמה הבאה (תמצא את הקוד בתקליטור המצורף לספר, קובץ בשם sessionID.asp):

```
<HTML>
<HEAD><TITLE> Session ID </TITLE></HEAD>
<BODY>
Your session ID is: <%=Session.SessionID %>
</BODY>
</HTML>
```

דף ASP זה פשוט מציג את ערכי המאפיין SessionID (ראה תרשים 5.2). כאשר משתמשים שונים מבקשים את דף זה יופיע מזהה session שונה בעבור כל אחד מהם. אבל, אם אותו משתמש יבקש את דף זה מספר פעמים תמיד יופיע אותו מזהה session.




**תרשים 5.2** הערך של SessionID

שימוש אחד למזהה ה-session הוא לשם מעקב אחר תנועת המבקרים באתר. למשל, תוכל לרשום לקובץ יומן (Log) את הדפים בהם ביקר משתמש כלשהו. כדי לעשות זאת עליך ליצור את הקובץ הבא ולכלול אותו בכל דף ודף:

```
<%
Who=Session.SessionID
CurrentPage=Request.ServerVariables("SCRIPT_NAME")
Response.AppendToLog Who&":"&CurrentPage
%>
```

תסריט זה משתמש בשיטה AppendToLog של האובייקט Request, כדי להוסיף רשומה לקובץ יומן השרת. בדוגמה זו, המחרוזת הנוספת ליומן השרת היא של מזהה ה-session, שאוחזר מהמאפיין SessionID. המחרוזת גם כוללת את הנתיב של הדף הנוכחי, אשר אוחזר ממשתנה השרת (Server Variable) בשם SCRIPT_NAME.

**הערה:**

באפשרותך להשתמש בשיטה AppendToLog של האובייקט Request כדי להוסיף כל מחרוזת של עד 80 תווים לקובץ היומן של שרת האינטרנט שלך. מאחר וסימני פסיק משמשים להפרדת שדות בקובץ היומן, אל לך להשתמש בסימנים אלה במחרוזות שלך.

קובץ היומן של Internet Information Server הוא קובץ טקסט המכיל רישום של פעולות שרת האינטרנט. תוכל לפתוח אותו באמצעות כל עורך טקסט רגיל. כברירת מחדל נמצא קובץ זה בתיקיה Winnt/system32/LogFiles.

# שליטה בסיום session

כיצד יודע השרת מתי מסתיים session? במילים אחרות, כיצד יודע השרת מתי המשתמש עזב את אתר האינטרנט שלך ועבר לאחר, או שהוא עזב את המחשב שלו והלך לסרט?

השרת מניח שאם מישהו לא ביקש או רענן דף כלשהו למשך זמן העולה על 20 דקות, אותו משתמש עזב, וה-session של אותו משתמש מסתיים. אסטרטגיית פעולה זו מאפשרת לשרת לפנות משאבים, בהם השתמש לצורך מעקב אחר session המשתמש.

ליישומי אתרי אינטרנט מסוימים משך הזמן 20 דקות קצר מדי. למשל, נניח שיש לך אתר משחקים הכולל מספר פאזלים מורכבים, ועל המשתמש לפתור אותם באמצעות נייר ועט. אתה עשוי להניח שהמשתמש יהיה לא פעיל, מנקודת מבטו של השרת, משך זמן ארוך.

ליישומי אתרי אינטרנט אחרים יהיה משך הזמן של 20 דקות ארוך מדי. אם יש לך אתר אינטרנט בו קיימת תנועה רבה של משתמשים, ואתה מעוניין להקל על העומס בשרת שלך, ייתכן ותבחר לקצר את אורך חייו של ה-session.

למרבה המזל, תוכל לשלוט במשך הזמן המירבי בו מורשה כל משתמש להיות לא פעיל (Inactive) לפני שה-session שלו יסתיים. לאובייקט Session יש מאפיין למטרה זו בדיוק. תוכל להגדיר את משך הזמן לפני סיום session באמצעות המאפיין Timeout של האובייקט Session. לדוגמה, התסריט הבא מגדיר את משך זמן חיי ה-session ל- 60 דקות:

```
<% Session.Timeout=60 %>
```

**טיפ:**

ניתן להגדיר את אורך חיי ה-session גם באמצעות Internet Service Manager. בתיבת הדו-שיח Application Configuration בחר בכרטיסיה Active Server Pages וקבע את מספר הדקות לאחריהן יסתיים ה-session (Session Timeout).

כאשר session משתמש מסתיים והוא מבצע בקשה חדשה, מתייחס השרת למשתמש כאל משתמש חדש. השרת יוצר session חדש וכל נתוני ה-session הישן אובד. אתה יכול לחייב זאת באופן ידני, על ידי השימוש בשיטה Abandon של האובייקט Session. ראה את התסריט הבא:

```
<HTML>
<HEAD><TITLE> Abandon Session </TITLE></HEAD>
<BODY>
<BR>The User Is <% Session.SessionID %>.
<% Session.Abandon %>
</BODY>
</HTML>
```

בדוגמה זו, מזהה ה-session של המשתמש מוצג על המסך, ואחר כך נקראת השיטה Session.Abandon. כאשר אתה מבקש דף חדש אובד כל המידע שבאובייקט Session ונוצר מזהה חדש המשויך ל-session. לאחר שנקראה השיטה Abandon מתייחס השרת למשתמש כאל משתמש חדש.

מצב אחד בו תמצא את השיטה Abandon יעילה הוא כאשר אתה מאחסן שם משתמש וסיסמה במשתנה session. תוכל להשתמש במשתני session אלה כדי לקבוע האם יש לאפשר למשתמש גישה לדף. אם אתה מעוניין לאפשר למשתמש לצאת מהמערכת, אתה יכול לספק לו קישור (Link) לדף בו נקראת השיטה Session.Abandon.

אבל עליך לזכור שהשפעתה של השיטה Abandon מופעלת רק לאחר שהדף הנוכחי מעובד עד תומו. משתני session משיגים את הערכים שלהם ממקום כלשהו בדף, אפילו מחלקים המופיעים אחרי השורה בה נקראת השיטה Abandon עצמה.

# sessions והקובץ Global.asa

בפרק 4 למדת להכיר את הקובץ Global.asa. כמו במקרה של יישום ASP, גם ל-session יש אירועים (Events). שניים מהם הם האירוע Session_OnStart המופעל כאשר session מתחיל, והאירוע Session_OnEnd אשר מופעל כאשר ה-session מסתיים.

השגרות המקבילות לאירועי session אלה חייבות להיות קיימות בקובץ Global.asa של היישום. להלן המבנה של הקובץ Global.asa, פעם נוספת:

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Application_OnStart
END SUB
</SCRIPT>

<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Application_OnEnd
END SUB
</SCRIPT>

<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Session_OnStart
END SUB
</SCRIPT>

<SCRIPT LANGUAGE=VBScript RUNAT=Server>
 SUB Session_OnEnd
 END SUB
</SCRIPT>
```

כדי ליצור תסריט המופעל בכל פעם שנוצר session חדש, עליך להוסיף את התסריט המיועד לחלק Session_OnStart שבקובץ Global.asa, ראה להלן בדוגמה הבאה:

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Session_OnStart
    Session("UserName")="Unknown"
    Session("")="Unknown"
END SUB
</SCRIPT>
```

תסריט זה מציב את הערך Unknown (לא ידוע) לשני משתני ה-session, UserName ו- UserPassword. דוגמה זו מדגימה את אחד מהשימושים העיקריים של התסריט Session_OnStart - איפוס משתני session.

תסריט Session_OnStart יכול לשמש למטרות אחרות גם כן. למשל, יישום מעניין של התסריט Session_OnStart הוא לשם הפניית מבקרים לדף חדש. נניח שאינך מעוניין שהמבקרים באתר שלך ייכנסו קודם כל לדף אחר כלשהו, אלא רק לדף הפתיחה של האתר. תוכל להפנות את הבקשה הראשונה של דף כלשהו לדף הפתיחה באמצעות השיטה Response.Redirect. הנה דוגמה:

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Session_OnStart
    MyHomePage=" /homepage.asp"
    RequestPage=Request.ServerVariables("SCRIPT_NAME")
IF NOT (STRCOMP(MyHomePage,RequestPage,vbTextCompare)=0) THEN
    Response.Redirect MyHomePage
END IF
END SUB
</SCRIPT>
```

**הערה:**

Microsoft משתמשת בשיטה זו כדי להפנות באופן אוטומטי משתמשים לדף הפתיחה של היישום לדוגמה Adventure Works, הנכלל בתוכנית ההתקנה של Option Pack עבור Windows NT.

בתסריט זה מתבצעת השוואה בין הנתיב שביקש המשתמש לבין נתיב דף הפתיחה של האתר. אם שני הנתיבים אינם זהים, המשתמש מופנה באופן אוטומטי לדף הפתיחה.

דוגמה אחרונה זו משתמשת הן בשיטה Session_OnStart והן בשיטה Session_OnEnd :

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Session_OnStart
    Response.AppendToLog Session.SessionID&" starting"
END SUB
</SCRIPT>
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
SUB Session_OnEnd
    Response.AppendToLog Session.SessionID&" ending"
END SUB
</SCRIPT>
```

התסריטים Session_OnStart ו-Session_OnEnd המופיעים כאן רושמים את מזהה ה-session של המשתמש לקובץ יומן. מכיון שהתסריט Session_OnStart מופעל מייד עם כניסתו של המשתמש, רושם התסריט את הזמן בו החל המשתמש session חדש. התסריט Session_OnEnd רושם מתי מסיים המשתמש את ה-session שלו, ועוזב את האתר. תוכל להיעזר במידע זה כדי לקבוע איזה דפים משמשים בדרך כלל כדי להיכנס או לצאת מאתר האינטרנט שלך.

תוכל ליצור אובייקטים עם טווח session בקובץ Global.asa. כדי לעשות זאת היעזר בתגית <OBJECT> המורחבת של Microsoft. הנה דוגמה היוצרת מופע של האובייקט Dictionary בעל טווח session :

```
<OBJECT RUNAT="Server" SCOPE="Session" ID="MyDictionary"
PROGID="Scripting.Dictionary"></OBJECT>
```

תכונת הטווח של התגית <OBJECT> מציינת שלמופע של האובייקט Dictionary הנקרא MyDictionary צריך להיות טווח session. לאחר שמופע של האובייקט Dictionary נוצר באופן זה, ניתן להתייחס אליו כפי שמתייחסים לאובייקט ASP מובנה. תוכל להשתמש בכל אחד מהמאפיינים והשיטות בכל דף ASP ביישום, מבלי שתצטרך להצהיר על האובייקט פעם נוספת.

**אזהרה:**

כאשר אתה מצהיר על אובייקט עם טווח session, היזהר ממודל בניית המטלות של האובייקט. הצהרה על מופע של אובייקט המשתמש במודל בניית מטלות על-פי דירות (Apartment-Threading Model), כמו האובייקט Dictionary, עלולה לגרום להשפעה קיצונית על ביצועי אתר אינטרנט בו יש תנועה רבה (ואף גרוע מכך, עלולות להיווצר גם שגיאות).

כאשר נוצר מופע של אובייקט עם טווח session, נוצר מופע חדש עבור כל משתמש. ככלל, הדבר אומר שאובייקטים שנוצרים עם טווח session דורשים כמות גדולה יותר של זיכרון מאשר אובייקטים עם טווח יישום. אבל, זה גם אומר שתתרחשנה פחות מריבות על השימוש באובייקט, מפני שרק משתמש אחד יכול לגשת לכל מופע של האובייקט.

# כיצד באמת פועלים ה-sessions

sessions משתמשים ב-Cookies (ראה בסעיף הבא פירוט רחב יותר אודות Cookies). כאשר משתמש מבקש בפעם הראשונה דף מהאתר שלך, יוצר השרת Cookie בודדת בדפדפן הלקוח, כדי לעקוב אחר ה-session. כאשר ה-session מסתיים פג גם תוקפה של ה-Cookie.

ה-Cookie הנוצרת עבור כל משתמש נקראת ASPSESSIONID (תוכל לצפות בו על ידי הצגת תוכן האוסף ServerVariables). המטרה היחידה של Cookie זו היא לספק מזהה ייחודי לכל משתמש.

משתני ה-session עצמם אינם מאוחסנים בדפדפן המשתמש, אבל ה-Cookie שנקראת ASPSESSIONID נדרשת כדי להשתמש במשתני session. השרת משתמש ב-ASPSESSIONID כדי לשייך את משתני ה-session המתאימים למשתמש הנכון. ללא ה-Cookie הזו, לא היתה לשרת דרך כלשהי לזהות את את אותו משתמש, כאשר הוא נע בין דפי האתר.

מזהה ה-session המאוחסן ב-ASPSESSIONID אינו זהה למאפיין SessionID. Microsoft משתמשת באלגוריתם מורכב כדי לחולל את הערך של ASPSESSIONID. היא עושה זאת כדי למנוע מפורצי מחשבים (האקרים, Hackers) לנחש את מזהה ה-session ולהתחזות למי שהם לא.

**הערה:**

ניתן לבטל sessions בשני אופנים: לבטל את ה- sessions לכל היישומים באתר שלך, או למנוע sessions מלהשתמש בדף מסוים.

כדי למנוע לחלוטין משרת האינטרנט ליצור sessions משתמשים, השתמש ב- Internet Service Manager. מתיבת הדו-שיח Application Configuration בחר בכרטיסיה Active Server Pages ובטל את הסימון ליד Enable Session State.

תוכל גם להגדיר כי דף ASP מסוים לא יעקוב אחר sessions משתמשים, וזאת על ידי שימוש בהפניית ASP הבאה, המוצבת בחלקו העליון של הקובץ:

```
<% EnableSessionState=False %>
```

מאחר והאובייקט Session משתמש ב-Cookies, הוא עלול להיות לא תואם לדפדפנים מיושנים או שאינם חדשים במיוחד. דפדפנים מיושנים פשוט אינם משתמשים ב-Cookies. מה שגרוע אף יותר הוא שדפדפנים לא מיושנים במיוחד, בגרסאות מתקדמות יחסית כגון גרסאות 4.0 של Netscape או Internet Explorer, מאפשרים לבטל את כל נושא ה-Cookies לגמרי.

דבר זה מציג בפניך בעיה. מכיון ש-Cookies אינן תואמות לכל סוגי הדפדפנים, עליך להיות זהיר בעת השימוש באובייקט Session כאשר אתה בונה את אתר האינטרנט שלך. למרות שקיימים דברים אותם פשוט אינך יכול לבצע מבלי להשתמש ב-sessions, ניתן לדמות מאפיינים מסוימים של ה-session באמצעים אחרים. אלטרנטיבות חלופיות לשימוש ב-Cookies ובאובייקט Session יידונו בסעיף מתקדם יותר של פרק זה, **שמירת מצב, ללא cookies**.

**הערה:**

דפדפנים מסוימים, כגון Netscape, משתמשים בכתובות URL תלויות רישיות (Case Sensitive URL) כדי לקבוע מתי לשלוח Cookie. דבר זה עלול ליצור בעיה כאשר לא נעשה שימוש ב-sessions. מסיבה זו ממליצה Microsoft שתמיד תשתמש באותו מבנה רישיות כאשר אתה מגדיר כתובות URL בדפי ASP. למשל, אל תשתמש בכתובת WWW/mypage.asp/ ובכתובת www/mypage.asp/, מפני שהדבר עלול ליצור בלבול עבור Netscape.

# עוגיות (Cookies)

מעט מאוד טכנולוגיות אינטרנט הצליחו להרגיז משתמשי אינטרנט רבים כל כך, כפי שהדבר עם Cookies. ל-Cookies יש שם תמים לכאורה, אבל משתמשים רבים מניחים שיש להן מטרות לא ממש תמימות. Netscape הציגה את ה-Cookies לעולם יחד עם הגירסה הראשונה של הדפדפן. מאז ועד היום איגוד ה-World Wide Web קבע תקן עבור Cookies, ורוב הדפדפנים העדכניים כבר תומכים בשימוש בהן.

מהן Cookies? דפדפנים התומכים ב-Cookies שומרים קובץ מיוחד אחד או יותר. קבצים אלה, הנקראים Cookie Files במחשבים מבוססי Windows ו- Magic Cookie File במחשבים מבוססי Macintosh, משמשים לאחסון מידע מאתרי אינטרנט. שרת אינטרנט יכול להעביר פיסות מידע לתוך קבצי ה-Cookie האלה. דבר זה מסביר את ההתנגדות הרבה שיש למשתמשים לאותם קבצים. אנשים אחדים מתייחסים ל-Cookies כאל חדירה לפרטיות שלהם. וגרוע אף מכך, אנשים מסוימים אף מתייחסים ל-Cookies כאל חדירה למרחב הפרטי שלהם.

קבצי Cookies מסוימים הם זמניים בלבד; אחרים קבועים יותר. למשל, תוקפן של ה-Cookies של ASP המשמשות למעקב אחר session משתמש פג מייד בתום ה-session, לאחר שהמשתמש עוזב את האתר. Cookies אחרות עשויות להישאר בקובץ Cookie הנשאר במחשב ואשר יכול להיקרא על ידי השרת אם וכאשר המשתמש שב לביקור חוזר באתר (תוכל להתבונן בתיקיה C:\Windows\Cookies).

אותן Cookies הנשארות במחשב, הן אלו הגורמות להסתייגות הגדולה. הפחד של המשתמשים הוא ש-Cookies אלו יכולות לשמש למעקב אחר הרגלי הגלישה של המשתמש. הדאגה היא שאם מידע זה נופל לידיים הלא נכונות, יכול המשתמש להפוך למטרה עבור מסעות הפרסום הנרחבים והמעצבנים באינטרנט.

גרסאות קודמות של Netscape ושל Internet Explorer כוללות אפשרויות נוספות המאפשרות שליטה טובה יותר על נושא ה-Cookies. תוכל להגדיר את הדפדפן שיקבל Cookies, או שיזהיר אותך לפני שה-Cookie מועברת אליך. בנוסף לכך, שני הדפדפנים מאפשרים למשתמש לבטל את אפשרות קבלת ה-Cookies כלל ועיקר.

כתוצאה מכל האמור, פותחו טכניקות מעניינות לביטול כל אפשרות חדירת Cookie למחשב, אפילו אם הדפדפן שלך אינו תומך באפשרות שכזו. למשל, כדי שלא לאפשר למחשב שלך לקבל Cookies מהאינטרנט תוכל לשנות את תכונת קבצי ה-Cookie לקריאה בלבד (לפרטים נוספים ראה באתר http://www.cookiecentral.com).

לרוע המזל, כפי שמצב הדברים כיום, אינך יכול לסמוך על Cookies כאשר אתה בונה את אתר האינטרנט שלך. Cookies יעשו את עבודתן עם מרבית הדפדפנים, אך לא עם הדפדפנים כולם, מה שיגרום לכישלון חרוץ של ה-session כולו.

**הערה:**

אתרי אינטרנט מסוימים שומרים מידע יקר ערך ב-Cookies שלהם. כדי לצפות במפרט ה-Cookies של Netscape פנה לאתר הבא:

http://home/netscape.com/newsref/std/cookie_spec.html

כדי ללמוד את מפרט איגוד ה- World Wide Web עבור Cookies, בקר באתר:
http://www.w3c.org.

לסיום, למידע כללי בנושא Cookies, פנה לכתובת:
http://www.cookiecentral.com.

# כיצד פועלות Cookies

Cookies מועברות בין הדפדפן והשרת (ובין השרת והדפדפן) באמצעות כותרות HTTP. השרת יוצר לראשונה את ה-Cookie על ידי שימוש בכותרת Set-Cookie בתגובה לבקשה. הבקשות הבאות מהדפדפן יחזירו את ה-Cookie הזו בכותרת Cookie.

נניח שאתה מעוניין ליצור Cookie בשם UserName המכילה את שמו של המבקר באתר שלך. כדי ליצור Cookie זו צריך השרת לשלוח כותרת כדוגמה הבאה:

```
Set-Cookie: UserName=BILL+Gates; path=/; domain=aspsite.com;
    expires=Tuesday, 01-Jan-05 00:00:01 GMT
```

כותרת זו מורה לדפדפן להוסיף רשומה בקובץ ה-Cookie שלו. הדפדפן מוסיף את ה-Cookie בשם UserName עם הערך Bill Gates. שים לב לכך שערך ה-Cookie מקודד בקידוד-URL.

מעבר לכך, הכותרת מורה לדפדפן ש-Cookie זו אמורה להיות מוחזרת לשרת, ללא קשר לנתיב שהוגדר בבקשה. אם מאפיין הנתיב היה מוגדר לערך אחר, כגון /private, ה-Cookie תוחזר רק בעת קבלת בקשות לנתיב זה. למשל, הבקשה לדף /private/mypage.htm תכלול את כותרת ה-Cookie, אך הבקשה לדף /mypage.htm לא.

המאפיין domain מגביל עוד יותר את המקום אליו יכולה להישלח Cookie על ידי הדפדפן. בדוגמה זו, יכולה ה-Cookie להישלח אך ורק לאתר www.aspsite.com. ה-Cookie לעולם לא תישלח לכתובת www.yahoo.com או לכל אתר אינטרנט אחר.

לסיום, המאפיין expires מציין מתי יפקע תוקפה של ה-Cookie. כותרת הדוגמה שלפנינו מורה לדפדפן לאחסן את ה-Cookie עד היום הראשון בינואר שנת 2005. בעצם, רוב הסיכויים שתוקף ה-Cookie יפוג הרבה קודם לכן, מכיון שכאשר גודלו של קובץ ה-Cookies גדל מעבר למידה מסוימת מתחיל הדפדפן למחוק ממנו Cookies, באופן אוטומטי.

לאחר שהדפדפן יצר Cookie הוא מחזיר אותה בכל בקשה שהוא מבצע מאתר האינטרנט - זאת אומרת, כל בקשה התואמת להגדרת הנתיב ב-Cookie. אבל, הדפדפן לא ישלח את ה-Cookie בבקשות שהוא מבצע לאתר אינטרנט אשר שם ה-Domain שלו שונה. הדפדפן ממשיך לשלוח את ה-Cookie עד שתוקפה פג. הכותרת Cookie נראית כך:

```
Cookie: UserName: Bill+Gates
```

# יצירה וקריאה של Cookies בדפי ASP

כדי ליצור Cookie בדפי ASP עליך להשתמש באוסף Cookies של האובייקט Response. ניתן ליצור שני סוגי Cookies: Cookie עם ערך בודד או Dictionary Cookie, המכילה מספר צמדי שמות וערכים.

ליצירת Cookie עם ערך בודד, תוכל להשתמש בתסריט הבא:

```
<%
Response.Cookie("UserName")="Bill Gates"
Response.Cookie("UserName").Expires="Jan 1, 2005"
%>
```

תסריט זה יוצר Cookie בשם UserName עם הערך Bill Gates. Cookie זו תוחזר על ידי דפדפן המשתמש עד לתאריך 1 בינואר שנת 2005, או עד שהדפדפן ימחק אותה. אם אינך מציין תאריך תוקף ל-Cookie, פג תוקפה כאשר המשתמש עוזב את האתר שלך.

מכיון שהתסריט הקצר אכן יוצר כותרת, עליך להציב את התסריט לפני כל הצהרות פלט בקובץ ASP שלך. לחילופין, תוכל לאגור את הדף (ראה את הסעיף **אגירת פלט הדף** בפרק 2).

התסריט שהוצג קודם לכן מהווה דוגמה פשוטה לאופן יצירת Cookie. הדוגמה משתמשת רק במאפיין Expires של האוסף Cookies. אבל, האוסף Cookies כולל מספר מאפיינים נוספים. הנה דוגמה מורכבת מעט יותר:

```
<%
Response.Cookie("UserName")="Steve Jobs"
Response.Cookie("UserName").Expires="Jan 1, 2005"
Response.Cookie("UserName").Path="/examples"
Response.Cookie("UserName").Domain=".aspsite.com"
Response.Cookie("UserName").Secure=True
%>
```

גם תסריט זה יוצר את ה-Cookie בשם UserName, אבל ל-Cookie הזו יש שלושה מאפיינים נוספים:

- המאפיין Path משמש לציון המקום אליו על הדפדפן לשלוח את ה-Cookie. בדוגמה זו תישלח ה-Cookie רק כאשר הנתיב לדף המבוקש מתחיל ב- examples/. למשל, ה-Cookie תישלח עם בקשה לדף examples/hello.asp/ או לדף examples/chapter5/cookies.asp/, אבל לא עם הבקשה לדף hello.asp/. כברירת מחדל משמש נתיב היישום.

- המאפיין Domain גם הוא מציין את המקום אליו תישלח ה-Cookie. בדוגמה הקודמת ה-Cookie תישלח עם בקשות ל-domain בשם aspsite.com. בלבד. הנקודה המקדימה מציינת שה-Cookie תישלח ל-www.aspsite.com, cricket.aspsite.com או beetle.aspsite.com (כל מחשב מארח, host, שב-domain בשם aspsite.com). אם מאפיין זה אינו מוגדר, התייחסות ברירת המחדל היא ל-domain של שרת האינטרנט.
- המאפיין Secure מציין שה-Cookie צריכה להישלח רק בשידור מוצפן. תוכל להשתמש במאפיין זה אם אתה משתמש בהצפנת Secure Layer Socket) SSL, ראה הרחבה בנושא בפרק 23).

כדי לקרוא Cookie מדף ASP עליך להשתמש באוסף Cookies של האובייקט Request. לדוגמה, כדי להציג ערך של Cookie תוכל להשתמש בתסריט הבא:

```
<%=Requset.Cookies("UserName") %>
```

תסריט זה מציג את הערך שב-Cookie בשם UserName. בדומה לכל האוספים בהם דנו עד כה, תוכל להיעזר במאפיין Count כדי לקבוע את מספר הפריטים באוסף Cookies. תוכל להשתמש בלולאה FOR...EACH או בלולאה FOR...NEXT כדי לעבור בין כל הפריטים שבאוסף. הנה דוגמה ללולאה FOR...EACH:

```
<%
FOR EACH thing IN Request.Cookies
    Response.Write("<BR>"&thing&"="&Request.Cookies(thing))
NEXT
%>
```

# יצירה של יותר מ-Cookie אחת

ניתן ליצור יותר מאשר Cookie אחת פשוט על ידי יצירת מספר Cookies עם האוסף Response.Cookies, כפי שהודגם בדוגמה הקודמת. אבל, דפדפנים רבים תומכים רק בשלוש-ארבע Cookies מאתר מסוים כלשהו.

קיימת גם שיטה חלופית ליצירת מספר Cookies. תוכל ליצור Cookies Dictionary. בעצם, Cookies Dictionary זו Cookie יחידה ובה מספר צמדי שמות וערכים. להלן דוגמה ליצירת Cookies Dictionary:

```
<%
Response.Cookie("User")("Name")="Bill Gates"
Response.Cookie("User")("Password")="billions"
%>
```

תסריט זה יוצר Cookies Dictionary עם השם User והמפתחות Name ו-Password. כאשר נוצרת Cookies Dictionary נשלחת לדפדפן כותרת כגון זו:

```
Set-Cookie:User=Name=Bill+Gates&Password=billions
```

נוצרת Cookie בשם User, אבל הערך של User הוא בעצם שני צמדים של שם וערך. השמות והערכים של כל מפתח ב-Cookies Dictionary מחוברים יחדיו ליצירת Cookie אחת גדולה.

כדי לאחזר Cookies Dictionary תוכל להשתמש באוסף Response.Cookies, כפי שהודגם קודם לכן. אם אתה פשוט מספק את שם ה-Cookies Dictionary, היא תוחזר באופן מלא, שאינו מדולל (Unsparsed). כדי לאחזר מפתחות מסוימים מתוך Cookies Dictionary עליך להעביר לאוסף את שם המפתח המבוקש. להלן דוגמה:

```
<%=Request.Cookie("User")%>
<%=Request.Cookie("User")("Name")%>
<%=Request.Cookie("User")("Password")%>
```

**אזהרה:**

כאשר שומרים ב- Cookie מידע מסווג, כגון סיסמאות, יש לנהוג במשנה זהירות. כברירת מחדל Cookies אינן מוצפנות בעת שהן משודרות בין הדפדפן ושרת האינטרנט, ולהיפך. אם אתה משתמש באבטחת SSL (Secure Sockets Layer, ראה בפרק 23) תוכל להשתמש בתכונה SECURE, כדי לשדר רק Cookies מוצפנות. אבל, בכל מקרה, ה-Cookie עצמה עדיין מאוחסנת כקובץ טקסט פשוט.

כדי לקבוע האם Cookie היא בעצם Cookies Dictionary, היעזר בתכונה HasKeys. למשל, התסריט הבא יחזיר את התשובה True אם ה-Cookie היא Cookies Dictionary. אחרת הוא יחזיר את התשובה False:

```
<%=Request.Cookie("User").HasKeys %>
```

# שמירת מצב, ללא Cookies

השימוש ב-sessions או ב-Cookies עלול להיות מסוכן, מפני שלא כל הדפדפנים תומכים בהם. מרגע שהתחלת להשתמש ב-Cookies באתר שלך אתה עשוי לקבל אין ספור תלונות ממבקרים באתר המשתמשים בדפדפנים שאינם מאפשרים את השימוש ב-Cookies. בסעיף זה תלמד מספר שיטות לשמירת המצב מבלי להשתמש ב-Cookies. במילים אחרות, תלמד כיצד לשמור מידע אודות משתמש מדף לדף. מוצגות כאן שלוש שיטות.

## שמירת מצב עם מחרוזות שאילתה

בפרק 2, הוסבר כיצד לעבוד עם מחרוזות שאילתה (Query strings). אתה יכול להוסיף מחרוזת שאילתה לכל קישור (Hyperlink) בדפי ASP שלך. על ידי השימוש במחרוזות שאילתה תוכל להעביר מידע מדף לדף, כמו בדוגמה הבאה:

```
<HTML>
<HEAD><TITLE> Query State </TITLE></HEAD>
<BODY>
<%
UserName=Server.URLEncode("Bill Gates")
%>
<A HREF="/nextpage.asp?<%=UserName%>">Click Here</A>
</BODY>
</HTML>
```

תסריט זה משייך את הערך Bill Gates למשתנה בשם UserName. כאשר המשתמש לוחץ על הקישור מועבר ערך משתנה זה לדף nextpage.asp במחרוזת שאילתה.

אתה יכול להמשיך ולהעביר את UserName מדף לדף, על ידי אחזור UserName מהאוסף QueryString. הדף nextpage.asp אליו עובר המשתמש בלחיצה על הקישור בדף הקודם עשוי להיראות דומה לזה:

```
<HTML>
<HEAD><TITLE> Next Page </TITLE></HEAD>
<BODY>
<%
UserName=Server.URLEncode(Request.QueryString("UserName"))
%>
<A HREF="/anotherpage.asp?<%=UserName%>">Click Here</A>
</BODY>
</HTML>
```

יתרונה של שיטה זו לאחזור מצב טמון בכך שהיא פועלת על כל הדפדפנים. אולם, יש להודות שהיא מגושמת מאוד. שהרי, אם אתה מעוניין לעקוב אחר המשתמש בכל דף ודף באתר שלך, עליך לכלול מחרוזת שאילתה עם כל קישור וקישור באתר. כל מחרוזת שאילתה חייבת לכלול את שם המשתמש.

חיסרון נוסף של שיטה זו לשמירת המצב נעוץ בכך שהיא אינה מאפשרת לך להעביר כמויות גדולות של נתונים. זכור, מחרוזות שאילתה אינן יכולות להיות גדולות במיוחד. כאשר גודל מחרוזת שאילתה עולה על 1,000 תווים (בערך), דפדפנים אחדים עשויים לקצץ אותה או שייכשלו בהפעלת הקישור כלל ועיקר.

## שמירת מצב עם שדות טופס נסתרים

אם אתה חייב להעביר כמות גדולה של נתונים מדף לדף, מבלי להשתמש במשתני session, לא עומדות בפניך ברירות רבות - עליך להשתמש בטופס HTML. אתה יכול להסתיר את המידע שאתה מעביר על ידי שימוש בשדות טופס נסתרים (Hidden form fields), כמו בדוגמה הבאה:

```
<HTML>
<HEAD><TITLE> Form State </TITLE></HEAD>
<BODY>
<%
UserName="Bill Gates"
%>
<FORM METHOD="Post" ACTION="/nextpage.asp">
<INPUT NAME="UserName" TYPE="HIDDEN" VALUE="<%=UserName"%>">
<INPUT TYPE="SUBMIT" VALUE="Next Page">
</FORM>
</BODY>
</HTML>
```

דף זה מכיל טופס HTML. לטופס יש שדה נסתר בשם UserName המכיל את ערכו של המשתנה UserName. הטופס כולל גם לחצן אחד. כאשר המשתמש לוחץ על לחצן זה נטען הדף הבא, nextpage.asp, והנתונים שבשדה הטופס הנסתר מועברים לדף החדש.

ניתן להמשיך ולהעביר נתונים באופן זה מדף לדף ללא סוף. בכל דף, עליך להשתמש באוסף Form של האובייקט Request כדי לאחזר את הנתונים שבשדה הנסתר. אחר כך, עליך ליצור שדה נסתר חדש, כך שהנתונים יכולים להיות מועברים שוב לדף אחר. הנה דוגמה לכך:

```
<HTML>
<HEAD><TITLE> Next Page </TITLE></HEAD>
<BODY>
<%
UserName=Request.Form("UserName")
%>
<FORM METHOD="Post" ACTION="/anotherpage.asp">
<INPUT NAME="UserName" TYPE="HIDDEN" VALUE="<%=UserName"%>">
<INPUT TYPE="SUBMIT" VALUE="Next Page">
</FORM>
</BODY>
</HTML>
```

## שילוב שיטות

אף אחת משתי השיטות שתוארו לעיל אינה אלגנטית במיוחד. אבל, אלו הן שתי החלופות היחידות לשמירת מצב, מבלי להשתמש במשתני session או Cookies. השימוש במחרוזות שאילתה ושדות טופס נסתרים משמר עבורך את התאימות עם כל סוגי הדפדפנים.

אם עליך לעקוב אחר משתמש בכל דף ודף באתר שלך, עליך לכלול מחרוזת שאילתה או שדה טופס נסתר בכל אחד מהדפים באתר. מרגע שהמשתמש לוחץ על קישור ערום (קישור שלא מצורפת אליו מחרוזת שאילתה) אינך יכול להמשיך ולעקוב אחריו.

**אזהרה:**

חיסרון בולט של השימוש במחרוזת שאילתה (Query string) או שדה טופס נסתר (Hidden form field) טמון בעובדה ששתי השיטות מחייבות העברת נתונים בין השרת והדפדפן, ובין הדפדפן והשרת. דבר זה מחייב אותך להישמר בעת העברת מידע מסווג, כגון סיסמאות.

תוכל לשלב את שתי השיטות לשמירת מצב. לדוגמה, בדפים מסוימים תוכל להשתמש במחרוזת שאילתה, כדי להעביר את שם המשתמש, ובדפים אחרים תעשה זאת על ידי שדה טופס נסתר. אם תפעל כך, לא תצטרך לבדוק גם את האוסף QueryString וגם את האוסף Form בכל דף ודף. אם אתה קורא לשיטה Request מבלי לציין שם של אוסף כלשהו, נבדקים שני האוספים באופן אוטומטי. ראה את הדוגמה הבאה:

```
<HTML>
<HEAD><TITLE> Next Page </TITLE></HEAD>
<BODY>
<%
UserName=Request("UserName")
%>
<FORM METHOD="Post" ACTION="/anotherpage.asp">
<INPUT NAME="UserName" TYPE="HIDDEN" VALUE="<%=UserName"%>">
<INPUT TYPE="SUBMIT" VALUE="Next Page">
</FORM>
<A HREF="/anotherpage.asp?<%=Server.URLEncode(UserName)%>">Click Here</A>
</BODY>
</HTML>
```

בדוגמה זו משויך למשתנה UserName שם המשתמש, ללא קשר לאופן העברת ערך המשתנה (בשדה טופס נסתר או במחרוזת שאילתה). הקריאה ל-("Request("UserName מאחזרת את הערך של UserName, בין אם הוא באוסף Form ובין אם הוא באוסף QueryString.

# סיכום

בפרק זה למדת כיצד לעבוד עם Sessions משתמש. למדת כיצד להשתמש באובייקט Session ליצירת משתני session (Session Variables) אשר יכולים לשמש לאחסון נתונים במספר רב של דפי אינטרנט. למדת כיצד ליצור תסריטים המופעלים כאשר session מתחיל וכאשר הוא מסתיים. בנוסף, למדת אודות קרובים של ה-sessions – כיצד לקרוא וליצור Cookies. לסיום, נסקרו מספר שיטות חלופיות לשמירת מצב, מבלי להשתמש ב-Cookies.

## פרק 6

# אינטראקציה עם הלקוח: ASP ו-Internet Explorer

**בפרק זה:**

- **אימות טופס בצד הלקוח**

פרק זה דן בנושא אימות טופס (תקינות המידע) בצד הלקוח, תוך שימוש ב-VBScript ו-JavaScript. תגלה בו כיצד למנוע טעויות צד-שרת על ידי אימות הנתונים שהוקלדו לטופס בצד הלקוח, לפני שהטופס נשלח. הדוגמאות יציגו אימות נתונים ומניעת טעויות חמורות. בנוסף, תלמד כיצד ליישם קובץ INCLUDE לשימוש חוזר בפעילויות אימות שכיחות.

## הערת העורך:

המונח **Validation** תורגם למילה **אימות** (מהמילה אמת) במשמעות של בדיקת תקינות הנתונים, לבדוק שהנתונים הגיוניים. למשל, תהליך האימות של מחרוזת תווים כלשהי, כדי לדעת אם היא מספר טלפון, יכלול (בין השאר) חיפוש אחר תווים שאינם ספרות (אסור שבמספר טלפון יימצאו התווים &, % ואחרים). בבדיקה זו לא נבדקת נכונות הנתונים. במקרה של מספר הטלפון: לא נבדק האם הטלפון הזה קיים ו/או המספר מחובר וכדומה. כוונת הבדיקה היא לוודא שהמספר הוא מספר טלפון חוקי (שאין לו, לדוגמה, רק 5 ספרות) ושהתוכנית בהמשך תוכל להתייחס אליו כמספר טלפון. אין מניעה שתהליך האימות (Validation) יכלול בדיקה מקיפה יותר שתבדוק מעבר לבדיקת תקינות בלבד. למשל, אם משתמש הקליד מספר טלפון סלולרי, תהליך האימות יכלול בדיקות למציאת תווים שאינן ספרות (דבר שאסור) וגם יבדוק שהקידומת היא בת 3 ספרות ושתי הספרות שמשמאל הן 05. כי אם הקידומת של הטלפון הסלולרי היא 09 זה לא קידומת של טלפון סלולרי.

# אימות טופס בצד הלקוח

אחת מההחלטות שתצטרך לקבל כמפתח ASP תהיה המקום בו יאומתו נתוני הטפסים. ברור שקיימים יתרונות לאימות בשרת ; למשל, תוכל להיות בטוח שהנתונים שלך מאומתים, ללא קשר לדפדפן בו משתמש הלקוח. אבל, כאשר אתה יודע שהלקוח משתמש בדפדפן עדכני, כגון Internet Explorer 5.0, תוכל לבצע את האימות בצד הלקוח. בין יתר הסיבות לביצוע אימות בצד הלקוח ניתן להדגיש את העובדה שאימות שכזה יפחית את העומס על השרת, וכך גם יפחית את משך ההמתנה של הלקוח, במידה וצריך לבצע תיקונים.

## אימות מבנה הנתונים בצד הלקוח

אחד מכאבי הראש הגדולים ביותר איתו צריכים להתמודד מפתחי ASP הוא שגיאה בבסיס הנתונים. את רוב השגיאות המבוססות על מבנה הנתונים, כגון ניסיון להקליד מחרוזת לשדה מספרי, ניתן למנוע מראש. בסעיף זה תלמד טכניקות רבות שיסייעו בידך לחזות ולטפל כמעט בכל אפשרויות שגיאות בסיס הנתונים הנגרמות על ידי הלקוח.

**טיפ:**

חלק משגרות האימות (validation routines) המוצגות בסעיף זה של הספר כתובות באופן מסורבל וארוך. ברוב המקרים תמצא שקל יותר, וגם יעיל בהרבה, לכלול קובץ נפרד אשר יכול להיקרא מתוך תסריט ASP, במקום לשוב ולחולל מחדש את הקובץ בכל תסריט הדורש אותו.

קובץ האימות validate.vbs, אותו תמצא בתיקיה Chapter06 שבתקליטור המצורף לספר זה, יכול להיכלל בשני אופנים :

כקובץ INCLUDE של ASP :

```
<SCRIPT LANGUAGE="VBScript">
<!-- #INCLUDE FILE="validate.vbs" -->
</SCRIPT>
```

או כקובץ VBScript מקושר :

```
<SCRIPT LANGUAGE="VBScript" SRC="validate.vbs">
</SCRIPT>
```

ההבדל בביצועים בין שתי אפשרויות אלו הוא זניח לחלוטין. אבל, אם אתה מעוניין לשפר את רמת האבטחה, רצוי שתתנסה בדרך השנייה המוצעת (כותב שאפתן בשפת VBScript יזדקק לצעד נוסף אחד לצפייה בקוד).

שגרות האימות הבאות יסייעו באימות מספר סוגי נתונים, כגון תאריכים, מספרי כרטיס אשראי, תאריכי תוקף כרטיס אשראי ואפילו כתובות דואר אלקטרוני ומבנה כתובות URL. כל שגרה תוצג בפניך בנפרד. אחר כך תראה כיצד תוכל להיעזר בשילוב כלשהו שלהן בעבור טופס מרובה שדות.

## isLength()

פונקציה זו מוודאת שאכן הוקלדו נתונים לשדה זה. היא עושה זאת על ידי השגת אורך ערך השדה באמצעות הפונקציה len() המובנית של VBScript. אם מוחזר האורך 0 (אפס) מותרע המשתמש כי לשדה חייב להיות ערך כלשהו. פונקציה זו מופיעה כאן ראשונה, מפני שהיא משמשת מספר פונקציות אחרות, במקומות בהם מחרוזת ריקה עלולה לגרום לשגיאות לא צפויות.

```
function isLength(val)
    if len(val.value)>0 then
        isLength = true
    else
        msgbox "This value can't be empty."
        val.select
        isLength = false
    end if
end function
```

## isNumber()

הפונקציה isNumber() בודקת אם ערך השדה הוא מספרי, תוך שימוש בפונקציה isNumeric() של VBScript. אם הנתונים בשדה זה הם משהו שאינו ערך מספרי מופיעה על המסך הודעה בתיבת הודעה מסוג MsgBox.

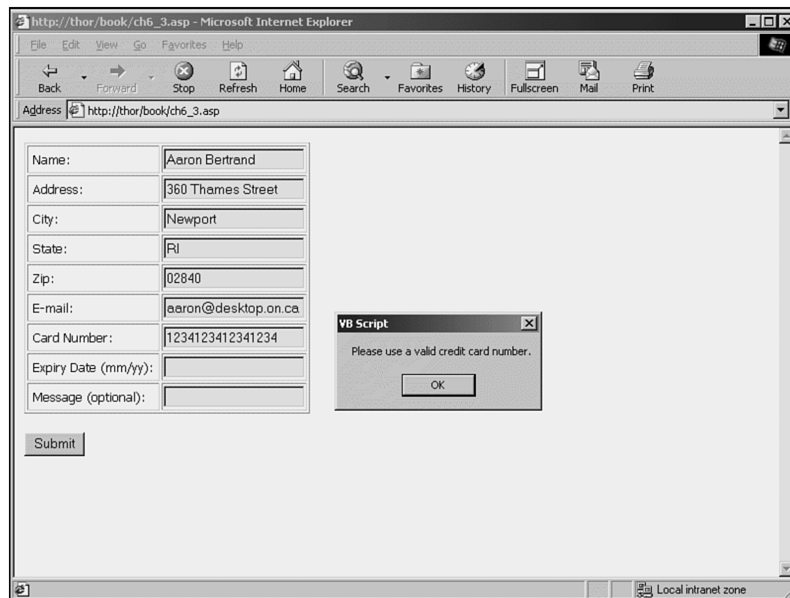
```
function isNumber(val)
    if isNumeric(val.value) then
        isNumber = true
    else
        msgbox "This value must be numeric."
        val.select
        isNumber = false
    end if
end function
```

## isCCnumber()

פונקציה זו נעזרת באלגוריתם בדיקת סכום לאימות שהשדה מכיל מספר אפשרי של כרטיס אשראי (למרות שבמקרים מיוחדים מאוד גם מספרים שגויים עשויים להיחשב כנכונים, כגון המספר 0000000000000000). שים לב לכך שפונקציה זו אינה בודקת את אמיתות וכשרות הכרטיס עצמו, וגם אינה בודקת אם הכרטיס אכן רשום על שם האדם הממלא את הטופס. פונקציה זו אומצה מקוד VB שנכתב על ידי ג'ון אנדרסון (John Anderson). את הפונקציה המקורית תוכל למצוא באתר האינטרנט שכתובתו http://www/planet-source-code.com/vb/scripts/ShowCode.asp?txtCodeId=902.

```
function isCCnumber(val)
    if isLength(val) then
        ccnumber = false
        val.value = trim(replace(val.value," ",""))
        if isNumeric(val.value) then
            ccnum = val.value
            for i = len(ccnum) to 2 step -2
                total = total + cint(mid(ccnum,i,1))
                tmp = cstr((mid(ccnum,i-1,1))*2)
                total = total + cint(left(tmp, 1))
                if len(tmp)>1 then
                    total = total + cint(right(tmp,1))
                end if
            next
            if len(ccnum) mod 2 = 1 then
                total = total + cint(left(ccnum,1))
            end if
            if total mod 10 = 0 then
                ccnumber = true
            end if
        end if
        if ccnumber then
            isCCnumber = true
        else
            msgbox "Please use a valid credit card number."
            val.select
            isCCnumber = false
        end if
    end if
end function
```

תוכל לראות קוד זה בפעולה כמוצג בתרשים 6.1.




**תרשים 6.1** הקלדת מספר לא חוקי של כרטיס אשראי

## isCCdate()

הפונקציה ()isCCdate נועדה להבטיח שערך השדה הוא תאריך פג תוקף חוקי, בין אם זה במבנה התאריך (mm/yy) ובין אם ביחס לחודש הנוכחי. פונקציה זו, הארוכה ביותר בין כל פונקציות האימות, פותרת שני נושאים מורכבים הקשורים בתאריכי כרטיס אשראי. האחד הוא הצורך לראות האם השנה הטבועה בכרטיס האשראי זהה לשנה הנוכחית; אם כן, עליך לבדוק שזהו לפחות החודש הנוכחי, או קדימה ממנו (אחרת מורה הדבר על כך שתוקף הכרטיס פג). הנושא השני הוא להציב שנים אשר נמצאות רחוק מדי בעתיד; זו היתה משימה קשה ביותר בשנים שקדמו לשנת 1999, מכיון ששנת 2000 גרמה לבעיות רבות. למרבה המזל, שנת 1999 כבר עברה וניתן לחשב כל שנה על ידי פעולת חיסור פשוטה יחסית (למשל, 2003-2000=3). בדוגמה הבאה אתה בודק עד 4 שנים קדימה (שזו התקופה המירבית בה תקף כרטיס אשראי כלשהו, בעיקר כרטיסים שמקורם בחו״ל. בארץ תוקף הכרטיס אינו עולה על שנתיים). תוכל לשנות את אורך התקופה על ידי שינוי שורה 17 בפונקציה זו (if yr<currYear+4 then).

```
function isCCdate(val)
    if isLength(val) then
        ccdate = false
        if len(val.value)>=3 and len(val.value)<=5 then
            if instr(val.value,"/")>0 then
              tCC = split(val.value,"/")
              if isNumeric(tCC(0)) and isNumeric(tCC(1)) then
                mn = cint(tCC(0))
                yr = cint(tCC(1))
                  currYear = cint(right(year(date()),2))
```

```
                if mn>0 and mn<13 then
                    if yr = currYear then
                        if mn >= month(date()) then
                            ccdate = true
                        end if
                    else
                        if yr<currYear+4 then
                            ccdate = true
                        end if
                    end if
                end if
            end if
        end if
    end if
    if ccdate then
        isCCdate = true
    else
        msgbox "Please enter a valid expiry date."
        val.select
        isCCdate = false
    end if
end if
end function
```

## isProperdate()

פונקציה זו בודקת שהערך שהוקלד הוא תאריך חוקי, תוך שימוש בפונקציה isDate() של VBScript. שים לב שהפונקציה אינה קשורה לנושא הגדרות אזוריות, כך שהתאריך 12/28/99, 28/12/00 ואפילו December 28, 2000 יתקבלו על ידה. אם התאריך שהוקלד אינו בפורמט תאריך מקובל מתבקש המשתמש לנסות פעם שנייה.

```
function isProperdate(val)
    if isDate(trim(val.value)) then
        isProperdate = true
    else
        msgbox "Please enter a valid date."
        val.select
        isProperdate = false
    end if
end function
```

## isPositive()

הפונקציה isPositive() משווה את ערך השדה לערך 0 (אפס). אם ערך השדה גדול מאפס מחזירה הפונקציה True, אחרת מתבקש המשתמש להקליד ערך חיובי. בדיקה זו מבוצעת, כמובן, לאחר שהערך נבדק ונמצא כערך מספרי.

```
function isPositive(val)
    if isNumber(val) then
        if val.value>=0 then
            isPositive = true
        else
            msgbox "Please enter a positive number."
            val.select
            isPositive = false
        end if
    end if
end function
```

## isNegative()

כמו הפונקציה ()isPositive, הפונקציה ()isNegative ראשית מוודאת שהערך שהועבר אליה הוא ערך מספרי. אם הוא אכן כזה, מושווה הערך לאפס. אם המספר אינו מספר שלילי, מתבקש המשתמש להקליד ערך שלילי.

```
function isNegative(val)
    if isNumber(val) then
        if val.value<0 then
            isNegative = true
        else
            msgbox "Please enter a negative number."
            val.select
            isNegative = false
        end if
    end if
end function
```

## isAlpha()

הפונקציה ()isAlpha בוחנת אם ערך השדה מכיל תווים ומספרים אלפא-נומריים; למשל, שהוא אינו מכיל תווים כגון מרכאות או גרש (שני התווים הללו יגרמו לתוהו ובוהו רציני בהצהרות SQL הנוצרות על ידי ASP). פונקציה זו נועדה להיות קלה יחסית להתאמה אישית, מפני שההגדרות שלך לערך אלפא-נומרי עשויות להיות שונות משל מפתח אחר. למשל, אתה עשוי לאפשר את התו @, בעוד שמפתח אחר אינו מעוניין לאפשר תו זה, אלא לאפשר את תווי הסוגריים. השורה השלישית מציגה את רשימת התווים האסורים כברירת מחדל; אם אתה מעוניין לאפשר תו כלשהו מרשימה זו, הסר אותו מהמשתנה isNot. שים לב שסימן המרכאות (") אינו נכלל בשורה השלישית, אלא רק בשורה החמישית. תו זה נבדק באמצעות הפונקציה המובנית ()instr והמקביל ה-ASCII שלו, chr(34). אם אתה מעוניין לאפשר מרכאות פשוט סמן את השורות חמש עד שבע (5-7) כהערה.

```
function isAlpha(val)
    if isLength(val) then
        isNot=" !@#$^*()_+=-'`~\|]}[{;:/?.,<>&%"
        invalid = false
        if instr(val.value,chr(34))>0 then
            invalid=true
        else
            for i = 1 to len(val.value)
                for x = 1 to len(isNot)
                    if mid(val.value,i,1)=mid(isNot,x,1) then
                        invalid = true
                    end if
                next
            next
        end if
        if not invalid then
            isAlpha = true
        else
            msgbox "Please use alphanumeric characters."
            val.select
            isAlpha = false
        end if
    end if
end function
```

## isEmail()

שמה של פונקציה זו עלול להטעות מעט. היא אינה בודקת שהכתובת אכן חוקית; היא רק מוודאת שהערך שהוקלד בשדה זה מכיל את התו @ ונקודה, וכי יש בו לפחות שישה תווים. הדרך היחידה להבטיח שהמשתמשים אכן מקלידים את כתובת הדואר האלקטרוני הנכונה שלהם היא לשלוח להם הודעת דואר אלקטרוני ובה בקשה לאישור, אותו יתבקש הלקוח, שקיבל את הדואר האלקטרוני, לשלוח בעת פתיחת הדואר. דבר זה אינו ניתן לביצוע באמצעות תסריט VBScript של צד-לקוח, והוא אינו חלק מהחומר הנלמד כאן.

```
function isEmail(val)
    if isLength(val) then
        e = val.value
        if instr(e,"@")>0 and instr(e,".")>0 and len(e)>5 then
            isEmail = true
        else
            msgbox "Please enter a proper email address."
            val.select
            isEmail = false
        end if
    end if
end function
```

## isZip()

הפונקציה ()isZip בודקת שקוד המיקוד שהוקלד לשדה זה הוא במבנה חוקי (בארץ אלו הן חמש ספרות, כגון 12345, ואילו בארה״ב יכולות להיות אף תשע ספרות המופרדות במקף, כגון 12345-6789). כמו במקרה של הפונקציה ()isEmail, גם פונקציה זו אינה בודקת את נכונות המיקוד, אלא רק את מבנהו החוקי. כדי לבחון אם המיקוד עצמו קיים וחוקי ניתן לרכוש בסיסי נתונים של חברות העוסקות בדבר, כגון דפי זהב או UPS (בארה״ב). אלה יאפשרו לך למפות את המיקוד לעיר, ואולי אף לשכונה.

```
function isZip(val)
    isZip = false
    tVal = trim(val.value)
    if len(tVal)=5 then
        if isNumeric(tVal) then
            isZip = true
        end if
    elseif len(tVal)=10 then
        l5 = left(tVal,5)
        r4 = right(tVal,4)
        m1 = mid(tVal,6,1)
        if isNumeric(l5) and isNumeric(r4) and m1="-" then
            isZip = true
        end if
    end if
    if not isZip then
        msgbox "Please enter a valid Zip code."
        val.select
    end if
end function
```

## isPath()

פונקציה זו יעילה כאשר המשתמשים שלך מתבקשים להקליד נתיב מקומי לקובץ או תיקיה. היא מוודאת שהנתיב שהוקלד מכיל את הגדרת הכונן (\:) ומשתמש במשתנה דומה ל-isNot, כדי למנוע מספר תווים שאינם אלפא-נומריים. התווים המורשים הם אלה השכיחים בנתיבים מקומיים, כגון קו תחתון, מקף או נקודה. כמו בפונקציה ()isAlpha, גם פונקציה זו ניתנת להתאמה, בהתאם לצרכיך. תוכל גם לקבוע אות כונן מסוימת, על ידי עריכת המחרוזת \: בשורה הרביעית כך שתהיה \:C, או כל אות כונן אחרת.

```
function isPath(val)
    if isLength(val) then
        path = false
        if instr(val.value,":\")>0 then
            isNot=" !@#$^*()'`~|]}[{;.>,<?%&+="
            path = true
            if instr(val.value,chr(34))>0 then
                path = false
```

```
            else
                for i = 1 to len(val.value)
                    for x = 1 to len(isNot)
                        if mid(val.value,i,1)=mid(isNot,x,1) then
                            path = false
                        end if
                    next
                next
            end if
        end if
        if path then
            isPath = true
        else
            msgbox "Please enter a valid local path."
            val.select
            isPath = false
        end if
    end if
end function
```

## isURL()

פונקציה זו משמשת כדי לאמת שהערך שהוקלד על ידי המשתמש הוא אכן כתובת URL. היא מאתרת את המחרוזת נקודתיים (:), קו נטוי (\), קו נטוי כפול (://) המוכר לנו כל כך (ואם תערוך את השורה הרביעית תוכל לקבוע שכתובת ה-URL תהיה http:// או ftp://). התווים המותרים כוללים סימן שאלה, סימני שוויון, סימן סולמית (hash), קו תחתון ונקודה. כמו פונקציות אחרות המשתמשות במחרוזת isNot, תוכל להכתיב איזה תווים מותרים, על ידי עריכת השורה החמישית של הפונקציה isURL().

```
function isURL(val)
    if isLength(val) then
        url = false
        if instr(val.value,"://")>0 then
            isNot=" !@$^*()'`~|]}[{;.>,<"
            url = true
            if instr(val.value,chr(34))>0 then
                url = false
            else
                for i = 1 to len(val.value)
                    for x = 1 to len(isNot)
                        if mid(val.value,i,1)=mid(isNot,x,1) then
                            url = false
                        end if
                    next
                next
            end if
        end if
```

```
        if url then
            isURL = true
        else
            msgbox "Please enter a valid URL."
            val.select
            isURL = false
        end if
    end if
end function
```

## ועכשיו, כולם ביחד

בקובץ ch6_3.asp תראה כיצד ניתן להשתמש בשילובים של פונקציות האימות (פונקציות הבודקות את תקינות הנתונים) שהוצגו כאן כדי לאמת בקלות יחסית את נתוניו של קובץ מרובה שדות. דוגמה זו בוחנת שמונה שדות בעיקר, תוך שימוש בפונקציות הבאות:

```
isLength()
isZip()
isEmail()
isCCnumber()
isCCdate()
```

שים לב לכך שהטופס בדף זה כולל בעצם תשעה שדות; השדה האחרון הוא אופציונלי. הוא נועד להדגים כיצד אתה יכול להשאיר שדות שלא יאומתו, במידה והם יכולים להיות ריקים, או להכיל כל סוג נתון או פורמט. בנוסף, שים לב שכאשר כל הנתונים שהוקלדו נכונים מוצגת תיבת הודעה (MsgBox), במקום לשלוח את הטופס. תדפיס 6.1 מציג את תוכן הקובץ ch6_3.asp.

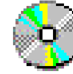
**תדפיס 6.1** שימוש בפונקציות אימות כלליות

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE=vbscript>
<!--#INCLUDE FILE="validate.vbs"-->
</SCRIPT>
</HEAD>

<BODY BGCOLOR=#ffffff>
<FORM NAME="f">
<TABLE BORDER=1>
    <TR><TD> Name: </TD>
        <TD><INPUT NAME="aname" TYPE="text"></TD></TR>
    <TR><TD> Address: </TD>
        <TD><INPUT NAME="address" TYPE="text"></TD></TR>
    <TR><TD> City: </TD>
        <TD><INPUT NAME="city" TYPE="text"></TD></TR>
    <TR><TD> State: </TD>
        <TD><INPUT NAME="state" TYPE="text"></TD></TR>
```

```
    <TR><TD> Zip: </TD>
        <TD><INPUT NAME="zip" TYPE="text"></TD></TR>
    <TR><TD> E-mail: </TD>
        <TD><INPUT NAME="email" TYPE="text"></TD></TR>
    <TR><TD> Card Number: </TD>
        <TD><INPUT NAME="ccnum" TYPE="text"></TD></TR>
    <TR><TD> Expiry Date (mm/yy): </TD>
        <TD><INPUT NAME="ccdate" TYPE="text"></TD></TR>
    <TR><TD> Message (optional): </TD>
        <TD><INPUT NAME="msg" TYPE="text"></TD></TR>
</TABLE>
<P>
<INPUT TYPE=button onClick='check()'
 VALUE="Submit">
 </FORM>
</BODY>
</HTML>

<SCRIPT LANGUAGE=vbscript>
function check()
    if not isLength(document.f.aname) then
        exit function
    elseif not isLength(document.f.address) then
        exit function
    elseif not isLength(document.f.city) then
        exit function
    elseif not isLength(document.f.state) then
        exit function
    elseif not isZip(document.f.zip) then
        exit function
    elseif not isEmail(document.f.email) then
        exit function
    elseif not isCCnumber(document.f.ccnum) then
        exit function
    elseif not isCCdate(document.f.ccdate) then
        exit function
    end if
    msgbox "Data is good; submit the form."
    'document.f.submit()
end function
</SCRIPT>
```

ברור שייתכן ויש פורמטים נוספים של נתונים עבורם תרצה לבנות פונקציות. ייתכן ותרצה לבדוק מספרי טלפון או קיצורים של מדינות (בארה״ב) או ערים (כגון ת״א או ראשל״צ). תוכל לעשות זאת תוך שימוש בקובץ validate.vbs כבסיס.

# אימות נתוני צד-שרת בצד הלקוח



סעיף זה מציג בפניך כיצד למנוע שגיאות בסיסי נתונים או שגיאות צד-שרת אחרות. הדבר נעשה על ידי בדיקת הנתונים עוד לפני שהטופס נשלח (לפני שממלא הטופס לוחץ על Submit). דוגמה מסוימת זו, אותה תמצא בקובץ ch6_4.asp בתקליטור, מונעת הוספת רשומה כפולה למסד נתונים של Access בו קיים שדה אינדקס המוגדר כ-no duplicates allowed. במקרה זה תשתמש בקובץ Chapter6.mdb פעם נוספת, ובטבלה Music. שדה האינדקס הוא Artist, כך שאתה תמנע מהוספת אותו אמן פעם נוספת. ללא אימות שכזה, הוספת רשומה כפולה של אותו אמן תגרום להופעתה של ההודעה הלא ממש נעימה הבאה:

[Microsoft][ODBC Microsoft Access 97 Driver] The changes you requested to the table were not successful because they would create duplicate values in the index, primary key, or relationship. Change the data in the field or fields that contain duplicate data, remove the index, or redefine the index to permit duplicate entries and try again.

כדי למנוע שגיאה כגון זו עליך למשוך את כל הערכים העשויים ליצור התנגשות ערכים לתוך שגרת אימות בצד הלקוח, תוך כדי הרצה. דבר זה ניתן לבצע על ידי יצירת לולאה על ערכת הרשומות (Recordset) ובניית שגרת If/Then עבור כל רשומה ייחודית. במאמץ קל תוכל למנוע מהמשתמש להיתקל בהודעות שגיאה של ODBC, ובמקום זאת הוא יקבל חיווי מיידי, עוד לפני שלחץ על Submit, לגבי כך שהוא מנסה להוסיף רשומה שכבר קיימת.

תדפיס 6.2 מראה כיצד הדבר נעשה.

**תדפיס 6.2** מניעת שגיאות ODBC מאימות בצד-לקוח

```
<HTML>
<HEAD>
<!--#INCLUDE FILE="db.inc"-->
<%
sql = "select artist from music"
set rs = conn.execute(sql)
%>
<SCRIPT LANGUAGE="JavaScript">
<!--
function check(frm) {
art = frm.art.value.toLowerCase();
art = art.replace("'","\'");
  if (art=='') {
   alert('Please enter a name.');
   frm.art.select();
   return false;
  }
<%
```

```
if not rs.eof then
  do while not rs.eof
    sart = replace(lcase(rs("artist")),"'","\'")
    response.write("   if (art=='" & sart & "') {" & chr(10))
    response.write("    alert('""" & sart & """ exists.\n")
    response.write("Please enter a different name.');")
    response.write(chr(10) & "    frm.art.select();")
    response.write(chr(10) & "    return false;")
    response.write(chr(10) & "   }" & chr(10))
    rs.movenext
  loop
end if
rs.close
set rs = nothing
conn.close
set conn = nothing
%>
   alert('submit()');
   // frm.submit();
}
//-->
</SCRIPT>
</HEAD>

<BODY BGCOLOR=#ffffff>
<FORM NAME="f1" onSubmit="check(document.f1);return false">
Enter an artist: <INPUT TYPE="text" NAME="art"><p>
<INPUT TYPE=submit VALUE="Check name >> ">
</FORM>
</BODY>
</HTML>
```

שים לב שהן בצד השרת והן בצד הלקוח, אתה ממיר את ערך האמן (Artist) לאותיות רגילות (Lowercase). דבר זה מונע את שאלת סוג האות; אם מישהו יקליד את השם כולו באותיות רישיות (Uppercase), או בתערובת של אותיות רישיות ורגילות, הן עדיין נחשבות לזהות, מפני שכולן כאמור מומרות לאותיות רגילות, לפני שנערכת ההשוואה. לדוגמה, כאשר תוקלד המילה Korn (בסדר אותיות כמו זה) היא תיחשב מייד כ-NOT EQUAL TO KORN, לא שווה ל-KORN. דבר נוסף שעליך לעשות הוא להקדים כל תו גרש (') במחרוזת יציאה (Escape Sequence, '\), כדי שכל גרש בשם כגון Stompin' Tom לא יתורגם בטעות על ידי תסריט צד-הלקוח כסימן סיום של מחרוזת (או, אם אתה משתמש ב-VBScript בלקוח, כתוחם פקודה, Command Delimiter). תדפיס 6.3 מציג חלק מהתסריט שהוא התוצאה.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="javascript">
<!--
function check(frm) {
art = frm.art.value.toLowerCase()
art = art.replace("'","\'");
    if (art=='korn') {
        alert('"korn" exists.\nPlease enter a different name.');
        frm.art.select();
        return.false;
    }
    if (art=='tiffany') {
        alert('"tiffany" exists.\nPlease enter a different name.');
        frm.art.select();
        return.false;
    }
    if (art=='  ') {
        alert('Please enter a name.');
        frm.art.select();
        return.false;
    }
    alert('submit()');
    // frm.submit();
}
// -->
</SCRIPT>
</HEAD>
<BODY BGCOLOR=#ffffff>
<FORM NAME="f1">
Enter an artist: <INPUT TYPE="text" NAME="art"><P>
<INPUT TYPE="button" onClick="check(document.f1);"
    value="Check Name >> ">
</FORM>
</BODY>
</HTML>
```

# סיכום

בדיקות אימות הן הכרח כדי שהתוכנית תפעל ותעשה את עבודתה כנדרש. מצד אחד זו טירחה ומצד שני זה חובה. למרות שהמשתמש התבקש להזין מספר שלם וחיובי הוא עשוי להזין ערכים אחרים (והכל בתום לב). אם התוכנית לא תדע לטפל במה שהזין המשתמש היא תעשה תציג תוצאות שגויות ו/או "תיזרק" החוצה.

# חלק 2

# רכיבים ברי-התקנה של דפי ASP

**בחלק זה:**

- **קבלת תוכן בצורה דינמית מהשרת**
- **עבודה עם מערכת הקבצים**
- **מעקב אחר תצוגת דפים והצגת פרסומות**
- **עבודה עם דואר אלקטרוני**
- **עבודה עם ממשקי Active Directory Service**

## פרק 7

# יצירת תוכן דינמי מהשרת

**בפרק זה:**

- **זיהוי מאפייני הדפדפן**
- **שימוש ברכיב Content Rotator**
- **רכיב Content Linking**

בפרק זה תלמד כיצד לעבוד עם שלושה רכיבי ActiveX שניתן להשתמש בהם בדפי ASP כדי ליצור תוכן דינמי. בחלק הראשון של הפרק תלמד כיצד להשתמש ברכיב **Browser Capabilities** כדי לזהות באופן אוטומטי את תכונות הדפדפן של המבקר באתר. בחלק השני תלמד כיצד להשתמש ברכיב **Content Rotator** להצגת תכני HTML שונים באופן אקראי. בחלק האחרון תלמד להשתמש ברכיב **Content Linking** כדי להוסיף באופן פשוט קישורים בין דפי ASP.

**הערה:**

הרכיב Browser Capabilities והרכיב Content Linking כלולים בחבילה Windows NT Option Pack. את הרכיב Content Rotator תוכל להוריד מאתר Microsoft בכתובת http://www.microsoft.com/iis, או לקנותו כחלק מה-Internet Information Server Resource Kit.

# מאפייני הדפדפן

אחד ההסברים לגידול העצום של רשת האינטרנט הוא פתיחות התקנים שלה. שפת HTML תוכננה כך שלא תהיה תלוית פלטפורמה או דפדפן. באופן תיאורטי, דף אינטרנט ייראה זהה בכל דפדפן ובכל מחשב. במציאות הסיפור שונה. כבר מההתחלה הציגה Netscape תגיות HTML המיוחדות לה. לדוגמה, הגירסה הראשונה של Netscape Navigator תמכה בתגית HTML להצגת טקסט מהבהב (התגית BLINK). עד היום, מרבית הדפדפנים שאינם של Netscape אינם תומכים בתגית הזו. **מסגרות** (Frames) הן דוגמה נוספת להרחבה של Netscape ל-HTML.

התחרות שהתפתחה בין Netscape לבין Microsoft רק החמירה את המצב. Microsoft הציגה מספר רב של תגיות HTML שפיתחה. לדוגמה, התגית <BGSOUND> המשמיעה קולות רקע, והתגית <MARQUEE> המציגה כתובת נגללת הוצגו לראשונה על ידי Microsoft, ורק דפדפני Microsoft תמכו בהן.

HTML הלכה והתפצלה בהתמדה למספר תקנים שונים. עם כל גירסה חדשה של דפדפני Netscape ו-Microsoft התרחב הפער בין מה שניתן לכנות "HTML של Netscape" לבין "HTML של Microsoft". ראוי לציין שמשתמשים רבים מעדיפים להשתמש בדפדפנים אופנתיים, אך איזוטריים, דוגמת Lynx או Opera. הדבר יוצר קושי רב למתכנני אתרי האינטרנט.

מצד אחד, מתכנן האתר מעוניין לכלול את התגיות החדשות ביותר, אך מצד שני הוא רוצה לשרת גם את המשתמשים בעלי הדרישות הייחודיות. לעולם אל תמעיט בערכו של "גורם החדשנות". אם ברצונך שמשתמשים ישובו לבקר באתר שלך, עליך לדחוף את HTML עד לקצה גבול היכולת שלו. עם זאת, עליך ליצור אתרים שיהיו נגישים לכל, באופן אוניברסלי. אף אחד אינו מעריך אתר שאין ביכולתו לראות. כשמשתמשים בתגיות HTML ייחודיות, מסתכנים באיבוד אוכלוסיות של משתמשים פוטנציאליים ייחודיים.

מה יעשה מתכנן דפי אינטרנט טוב כדי לתמרן בין שני לחצים סותרים אלה? התשובה של Microsoft היא להשתמש ברכיב Browser Capabilities, כמתואר בסעיף הבא.

## שימוש ברכיב Browser Capabilities

תוכל להשתמש ברכיב **Browser Capabilities** כדי להציג דפי אינטרנט שונים, לפי יכולות הדפדפן. לדוגמה, דפדפנים מסוימים תומכים במסגרות (Frames), ואחרים לא. על ידי שימוש ברכיב Browser Capabilities תוכל לגלות אם הדפדפן תומך בהצגת דפי אינטרנט במסגרות, ולכן גם תוכל להציג גירסה של האתר בה יש שימוש במסגרות רק לדפדפנים התומכים בכך.

**הערה:**

ייתכן שרכיב Browser Capabilities יזהה תכונות רבות יותר מאלו המוצגות כאן. הדבר תלוי בגרסת הקובץ browscap.ini המותקנת במחשב שלך. לפרטים נוספים עיין בהמשך תחת הכותרת "**כיצד באמת פועל הרכיב Browser Capabilities**".

**הערה:**

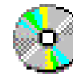
תוכל להבחין בתכונות מסוימות של הדפדפן או המחשב, דוגמת רזולוציית המסך או עומק הצבע, על ידי שימוש באוסף (Collection) המשתנים ServerVariables. הדבר פועל רק עם גרסאות מסוימות של Microsoft Internet Explorer. מידע נוסף ראה בפרק 2.

כברירת מחדל, הרכיב Browser Capabilities יזהה את תכונות הדפדפן הבאות:

- browser - סוג הדפדפן. לדוגמה, Internet Explorer או Netscape.
- version - מספר הגירסה המלא של הדפדפן.
- majorver - מספר הגירסה הראשי של הדפדפן (המספר שלפני הנקודה).
- minorver - מספר הגירסה המשני של הדפדפן (המספר שאחרי הנקודה).
- frames - מציין אם הדפדפן תומך במסגרות.
- tables - מציין אם הדפדפן תומך בטבלאות.
- cookies - מציין אם הדפדפן תומך ב-Cookies.
- backgroundsounds - מציין אם הדפדפן תומך בתגית <BGSOUND>.

- vbscript - מציין אם הדפדפן תומך בתסריטי צד-לקוח מסוג VBScript.
- javascript - מציין אם הדפדפן תומך בתסריטי צד-לקוח מסוג JavaScript.
- javaapplets - מציין אם הדפדפן תומך ביישומוני Java.
- activexcontrols - מציין אם הדפדפן תומך בפקדי ActiveX של צד-לקוח.
- beta - מציין אם הדפדפן נמצא עדיין בגרסת בטא שלו (גרסת ניסוי).
- platform - מציין את מערכת ההפעלה של הדפדפן. לדוגמה, Windows 95, Windows NT או MacOS PowerPC.
- Win16 - מציין אם הדפדפן פועל בסביבת Windows 3.x במקום Windows 9x או Windows NT.

בניגוד לאובייקטים שנידונו בפרקים הקודמים, דוגמת האובייקטים Request וResponse, הרכיב Browser Capabilities הוא רכיב בר-התקנה. פירוש הדבר שיהיה עליך ליצור מופע (Instance) בעל טווח (Scope) ייחודי של הרכיב לפני שתוכל להשתמש בו (למידע נוסף על יצירת רכיבים בעלי טווחים שונים, ראה פרק 1).



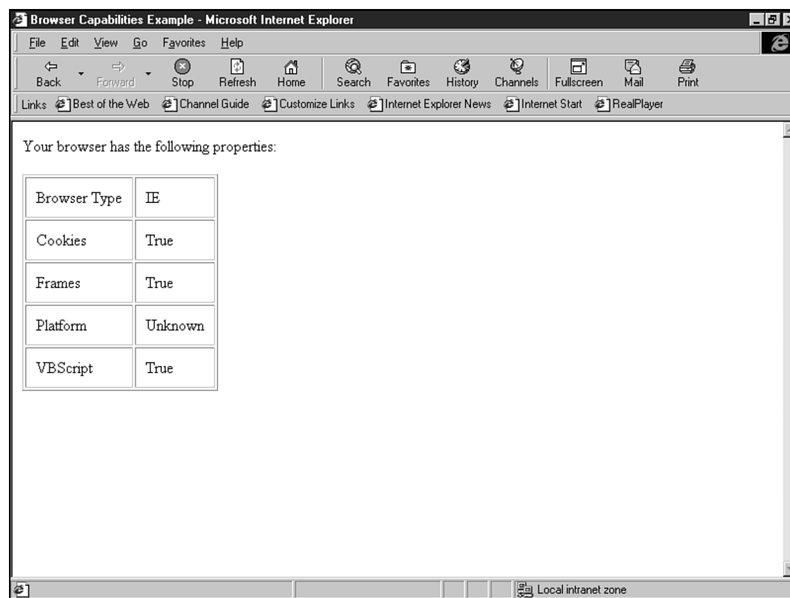
לאחר שתיצור מופע של הרכיב, תוכל לצרף את שם התכונה שברצונך לגלות אל שם מופע הרכיב. לדוגמה, דף ASP שבתדפיס 7.1 מציג מספר תכונות של הדפדפן. (תמצא את הקובץ browserFeatures.asp בתקליטור המצורף לספר זה, בתיקיה Books\59303\Chap07).

**תדפיס 7.1** הצגת מאפייני דפדפן

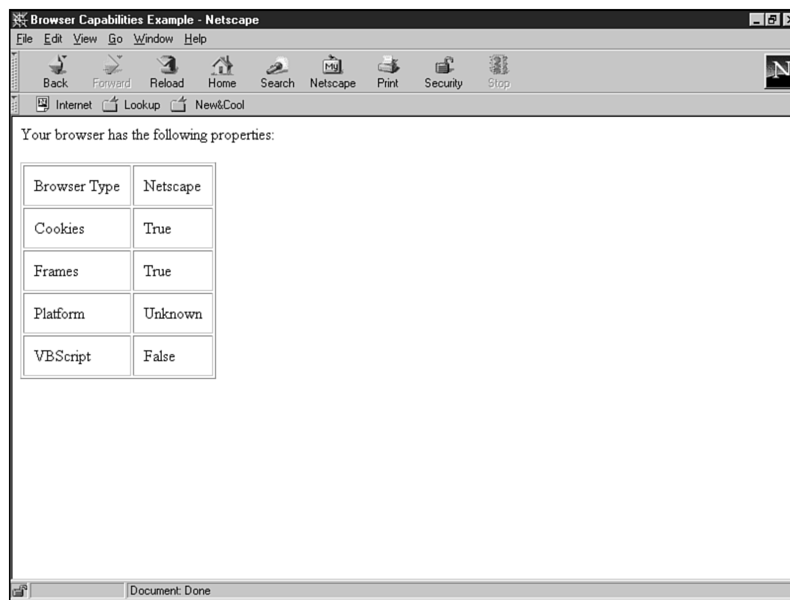
```
<HTML>
<HEAD><TITLE> Browser Capabilities Example </TITLE></HEAD>
<BODY>
<%
Set MyBrow=Server.CreateObject("MSWC.BrowserType")
%>
Your browser has the following properties:
<P>
<TABLE BORDER=1 CELLPADDING=10>
<TR>
<TD>Browser Type</TD><TD><%=MyBrow.Browser%></TD>
</TR>
<TR>
<TD>Cookies</TD><TD><%=MyBrow.Cookies%></TD>
</TR>
```

```
<TR>
<TD>Frames</TD><TD><%=MyBrow.Frames%></TD>
</TR>
<TR>
<TD>Platform</TD><TD><%=MyBrow.Platform%></TD>
</TR>
<TR>
<TD>VBScript</TD><TD><%=MyBrow.vbscript%></TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

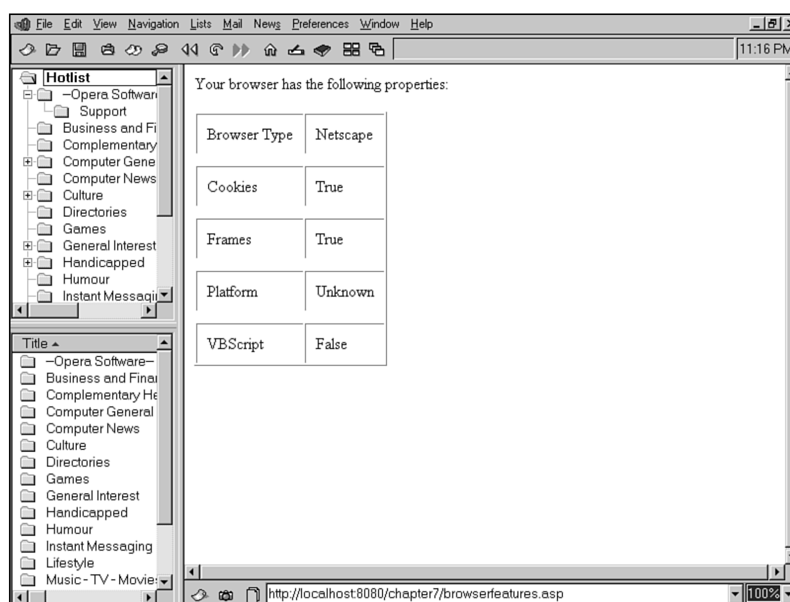
תסריט זה מגלה ומציג מספר תכונות דפדפן. תוצאות שונות יוצגו לדפדפנים שונים. תרשים 7.1 מציג דוגמת פלט עבור שימוש בדפדפן Internet Explorer 4.0, תרשים 7.2 מציג דוגמת פלט עבור שימוש ב- Netscape Navigator 4.5, ולבסוף, תרשים 7.3 מציג דוגמת פלט עבור שימוש ב- Opera 3.51.




**תרשים 7.1** דוגמה לפלט רכיב Browser Capabilities בעת שימוש ב-Internet Explorer 4.0.




**תרשים 7.2** דוגמה לפלט רכיב Browser Capabilities בעת שימוש ב-Netscape Navigator 4.5.




**תרשים 7.3** דוגמה לפלט רכיב Browser Capabilities בעת שימוש ב-opera 3.51.

## כיצד פועל הרכיב Browser Capabilities

חשוב להבין כיצד באמת פועל הרכיב Browser Capabilities, כדי להבין כמה מגבלות מהותיות שלו. הרכיב מזהה תכונות של דפדפן על ידי שימוש בכותרת HTTP Request (מידע הנשלח מהדפדפן לשרת בזמן התקשורת בינהם) ועל ידי שימוש בקובץ טקסט מיוחד המכיל מידע אודות הדפדפן.

בכל פעם שדפדפן יוצר בקשה, הוא כולל בה כותרת מסוג USER-AGENT. כותרת זו כוללת מידע אודות סוג הדפדפן שבשימוש, ומספר הגירסה שלו. תוכל לאחזר כותרת זו ישירות על ידי שימוש באוסף ServerVariables של האובייקט Request, כך :

```
<%=Request.ServerVariables("HTTP_USER_AGENT")%>
```

בעת שימוש ב- Netscape Navigator 3.0 עבור Windows NT לדוגמה, הערך שיוחזר יהיה :

```
Mozilla/3.0 (WinNT; 1)
```

חשוב להבין כי זהו המידע היחיד המועבר בין הדפדפן לשרת. הרכיב Browser Capabilities מסתמך על המידע שבכותרת זו. הרכיב אינו מזהה כל תכונה של הדפדפן באופן ישיר.

הרכיב Browser Capabilities מאחזר (קורא) את ערך ה-header של USER-AGENT header ומנסה לשייך אותה להגדרות דפדפן בקובץ מיוחד בשם browscap.ini. הקובץ browscap.ini נמצא בשרת, בדרך כלל בתיקיה Winnt/System32/inetsrv/. בעת התקנת דפי ASP, קובץ זה מותקן גם הוא באופן אוטומטי.

הקובץ browscap.ini הוא קובץ טקסט רגיל, אשר מכיל רשימה של דפדפנים ותכונותיהם. לדוגמה, לפניך הגדרת הדפדפן Netscape Navigator גירסה 2.0 בקובץ browscap.ini :

```
[Netscape 2.0]
browser=Netscape
version=2.0
majorver=2
minorver=0
frames=TRUE
tables=TRUE
cookies=TRUE
backgroundsounds=FALSE
vbscript=FALSE
javascript=TRUE
javaapplets=TRUE
beta=FALSE
Win16=FALSE
```

כשהרכיב Browser Capabilities מדווח על תכונות הדפדפן Netscape Navigator 2.0 הוא מסתמך על הגדרות אלו.באפשרותך לשנות טקסט זה באופן ישיר. למשל, על ידי שינוי הערך של backgroundsound מ-FALSE ל-TRUE תוכל לציין, בשוגג כמובן, כי הדפדפן Netscape Navigator 2.0 תומך בתגית <BGSOUND>.

קיימות מספר תכונות חשובות של דפדפנים שאין באפשרות הרכיב Browser Capabilities לזהות. לדוגמה, יהיה שימושי מאד לדעת אם דפדפן מסוים תומך ב-Secure Sockets Layer או ב-Cascading Style Sheets. מכיון שמידע זה אינו נכלל בקובץ browscap.ini, לא תוכל להשתמש ברכיב Browser Capabilities לזיהוי תכונות אלו. אבל, תוכל להוסיף מידע זה לקובץ browscap.ini בעצמך. לדוגמה, תוכל להוסיף את שתי השורות הבאות להגדרות הערך של Netscape Navigator 2.0:

```
SSL=TRUE
CSS=FALSE
```

לאחר הוספת שורות אלו, הרכיב Browser Capabilities ידווח על התכונות עבור Netscape Navigator 2.0. בכל פעם שהרכיב יזהה את הדפדפן הזה, הוא יניח שהדפדפן מכיל מאפיינים אלה. לדוגמה, התסריט הבא יחזיר TRUE אם תכלול אותו בדף המאוחזר על ידי הדפדפן Netscape Navigator 2.0:

```
<%MyBrow.SSL %>
```

ייתכן שהבחנת כי רבות מהגדרות הדפדפנים בקובץ browscap.ini נראות כך:

```
[Mozilla/2.0 (Win95; U)]
parent=Netscape 2.0
platform=Win95
```

כשהגדרת דפדפן כוללת את הפרמטר parent (הורה), ההגדרה יורשת את כל תכונות ההורה שלה. ההגדרה הקודמת יורשת את כל התכונות של הדפדפן Netscape Navigator 2.0. לדוגמה, למרות שההגדרה לא מציינת האם גרסת Netscape למערכת Windows 95 תומכת במסגרות, ידווח הרכיב Browser Capabilities כי גירסה זו תומכת בהן, מכיון שהן נתמכות על ידי ההורה.

הודות לפרמטר parent, אין צורך לציין את אותה האינפורמציה שוב ושוב. ניתן ליצור הגדרת הורה אחת (Parent definition), ולאחר מכן לרשת ממנה מספר הגדרות-בת (Child definitions) שיכילו מידע ספציפי יותר. במקרה של התנגשות בין הגדרות-הבת להגדרות-ההורה, תינתן עדיפות להגדרות-הבת.

דיוקו של הרכיב Browser Capabilities תלוי בדיוק הקובץ browscap.ini. במידה ומישהו ישתמש בדפדפן או בגרסת דפדפן שלא נמצאות עבורו הגדרות בקובץ, לא יוכל הרכיב Browser Capabilities לדווח עבורו את התכונות באופן מדויק. כשאין באפשרות הרכיב Browser Capabilities לזהות דפדפן, ידווחו בעבור דפדפן זה התכונות המוגדרות עבור **דפדפן ברירת המחדל** (Default Browser). להלן דוגמה להגדרת דפדפן ברירת מחדל:

```
[Default Browser Capability Setting]
browser=Default
version=0.0
majorver=#0
minorver=#0
frames=FALSE
tables=TRUE
cookies=FALSE
backgroundsounds=FALSE
vbscript=FALSE
javascript=FALSE
javaapplets=FALSE
activexcontrols=FALSE
AK=FALSE
SK=FALSE
AOL=FALSE
beta=FALSE
Win16=FALSE
Crawler=FALSE
CDF=FALSE
```

שוב, במידה ואינך מרוצה ממאפייני ברירת המחדל המצוינים בקובץ browscap.ini, ביכולתך לשנותם ישירות בקובץ. לדוגמה, ייתכן ותרצה להניח כי לא כל הדפדפנים מסוגלים להציג טבלאות. כדי לשנות הנחה זו, כל שעליך לעשות הוא לשנות את ערך המאפיין tables בהגדרת דפדפן ברירת המחדל.

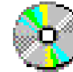
**הערה:**

כדאי לחפש עדכונים לקובץ browscap.ini באתר של Microsoft, שכן החברה מציגה עדכונים תכופים של קובץ זה. בדוק באתר http://www.microsoft.com/iis.

מספר גופי צד-שלישי מציגים עדכונים לקובץ גם הם. דוגמה אחת לכך היא האתר BrowsCap Central, בכתובת http://www.cyscape.com/asp/browscap/.

# יישום לדוגמה של הרכיב Browser Capabilities

חלק זה מציג דוגמה ליישום העושה שימוש ברכיב Browser Capabilities. מלבד אופן השימוש ברכיב, ניתן למצוא כאן דוגמה גם למגבלותיו.



בדף ASP המוצג בתרשים 7.2 נעשה שימוש ברכיב Browser Capabilities כדי לזהות אם הדפדפן תומך בהצגת מסגרות. אם הדפדפן תומך במסגרות, מוצגת גירסה ממוסגרת של הדף. אם הדפדפן לא תומך בהן מוצג דף המודיע כי הצפיה באתר דורשת דפדפן התומך במסגרות. (תמצא את הקובץ checkFrames.asp בתקליטור המצורף לספר זה, בתיקיה Books\59303\Chap07).

**תדפיס 7.2** זיהוי תמיכה במסגרות

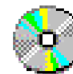
```
<%
Set MyBrow=Server.CreateObject("MSWC.BrowserType")
IF MyBrow.Frames THEN
%>
<HTML>
<HEAD><TITLE> Framed Page </TITLE></HEAD>
<FRAMESET COLS="100,*">
<FRAME SRC="menu.asp">
<FRAME SRC="body.asp">
</FRAMESET>
</HTML>
<% ELSE %>
<HTML>
<HEAD><TITLE> Frameless Page </TITLE></HEAD>
<BODY>
We have detected that your browser is incapable of using frames.
You are using a <%=MyBrow.browser%> browser
(version <%=MyBrow.version %>).
To download a more recent browser, please visit:
<P> <A HREF="http://www.microsoft.com">Microsoft</A>
<P> OR
<P> <A HREF="http://www.netscape.com">Netscape</A>
</BODY>
</HTML>
<% END IF %>
```

דף ASP זה מתוכנן להציג שני דפים שונים, על פי תנאי. אם הרכיב Browser Capabilities זיהה כי דפדפן המשתמש מסוגל לפרש את תגיות המסגרות, מוצג הדף הראשון. דף זה מציג שתי מסגרות. אחרת, מוצג הדף שני. שים לב כיצד משמש הרכיב גם לדיווח שם וגרסת הדפדפן שבשימוש.

דוגמה זו עושה יותר מלהמחיש את אופן השימוש ברכיב Browser Capabilities. היא גם מבהירה מדוע לא כדאי בדרך כלל להשתמש ברכיב למטרה זו. היא מציגה בעיה רצינית ברכיב Browser Capabilities.

הבעיה בדף זה היא שהדף השני יוצג בכל מקרה בו ייתקל דף ה-ASP בדפדפן שהוא אינו מזהה. במילים אחרות, גם אם הדפדפן הלא מזוהה כן תומך בהצגת מסגרות, יוצג הדף השני, שכן ההגדרות בקובץ browscap.ini קובעות כי כברירת מחדל, דפדפנים לא מסוגלים להציג מסגרות.

הפגם הראשי של הרכיב Browser Capabilities נעוץ בעובדה שעליו להסתמך על המידע שהוצב על ידי Microsoft או על ידך בקובץ browscap.ini. הקצב המהיר בו טכנולוגיות ודפדפנים חדשים מוצגים/ות ברשת האינטרנט מערער את הערך של רכיב זה.



כשהדבר אפשרי, עדיף להשתמש ב- HTML כדי להציג תוכן המשתנה לפי יכולות הדפדפן. לדוגמה, דרך טובה יותר לזהות האם הדפדפן תומך במסגרות היא שימוש בתגית HTML התקנית <NOFRAMES>, כמוצג בתדפיס 7.3 (ראה קובץ checkFramesHTML.asp).

**תדפיס 7.3** זיהוי תמיכה במסגרות על ידי שימוש בקוד HTML

```
<HTML>
<HEAD><TITLE> Framed Page </TITLE></HEAD>
<FRAMESET COLS="100,*">
  <FRAME SRC="menu.asp">
  <FRAME SRC="body.asp">
<NOFRAMES>
  We have detected that your browser is incapable of using frames.
  You are using a <%=MyBrow.browser%> browser
  (version <%=MyBrow.version %>).
  To download a more recent browser, please visit:
  <P> <A HREF="http://www.microsoft.com">Microsoft</A>
  <P> OR
  <P> <A HREF="http://www.netscape.com">Netscape</A>
</NOFRAMES>
</FRAMESET>
</HTML>
```

לשיטה זו יתרון כפול. ראשית, השימוש בתגית <NOFRAMES> עונה על מפרט WWWC (World Wide Web Consortium) לגבי HTML 4.0. WWWC הוא ארגון המופקד על ניהול תקן קבוע עבור HTML (ראה http://www.w3.org). יתרונה השני של שיטה זו הוא בכך שתעבוד עם כל דפדפן שהוא, ישן, חדש או איזוטרי ככל שיהיה.

אם הדפדפן לא תומך במסגרות, תוצג גרסת האתר הלא ממוסגרת - גירסה זו נמצאת בין התגיות <NOFRAMES>. הדבר יעבוד גם אם הדפדפן לא תומך בתגית <NOFRAMES> באופן מפורש. כשדפדפן לא מזהה תגית HTML, הוא פשוט מתעלם ממנה. הדפדפן יתעלם מכל התגיות הקשורות למסגרות, ויציג במקומן את התוכן הלא ממוסגר בלבד.

# שימוש ברכיב Content Rotator

הרכיב **Contenet Rotator** משמש להצגת תכני HTML שונים באופן אקראי. להלן מספר רעיונות אפשריים לשימוש ברכיב:

- **טיפ יומי** - ניתן להשתמש ברכיב להצגת טיפים שונים לשימוש באתר, כמו *"בקרו בפורום שלנו אם ברצונכם לשוחח עם משתמשים חדשים"*.
- **מבזק חדשות** - הרכיב יכול לשמש למעבר על רשימה של אירועים חדשים, לדוגמה - *"המדריך השלם ASP 3 בהוצאת הוד עמי - עכשיו בחנויות!"*.
- **קישור אקראי** - תוכל להשתמש ברכיב להצגת קישור אקראי לאתר מתוך רשימת האתרים החביבים עליך.
- **כרזות פרסום** (Banner Advertisement) - בדומה לרכיב Ad Rotator (המוזכר בהרחבה בפרק 9), ניתן להשתמש ברכיב זה להצגת כרזות פרסום. השימוש ברכיב זה ייתן לך גמישות רבה יותר באופן הצגת הפרסומות. לדוגמה, תוכל להשתמש ברכיב להצגת פרסומות עשירות מדיה (פרסומות העושות שימוש בקוד HTML דינמי).
- הצגה אקראית - ניתן להציג שם מוצר מבין רשימת המוצרים שבאתר, בצורה אקראית.

**הערה:**

רכיב זה לא נתמך על ידי Microsoft באופן רשמי, ואינו כלול בחבילת האופציות של Windows NT, כך שייתכן ותתקשה בהשגתו. מפעם לפעם, מושם הרכיב להורדה באתר של Microsoft (http://www.microsoft.com/iis). אם אינך מצליח למצוא שם, לא תהיה לך ברירה, אלא לרכוש את Microsoft Internet Information Resource Kit.

כדי להשתמש ברכיב זה להצגת מחרוזת של תוכן HTML, יש להשתמש בשיטה ()ChooseContent של הרכיב. שיטה זו מאחזרת מחרוזת HTML מתוך קובץ מיוחד הנקרא **קובץ תזמון התוכן** (content schedule file), ומציגה אותה בדף ASP, כפי שמוצג בדוגמה הבאה:

```
<HTML>
<HEAD><TITLE> Home Page </TITLE></HEAD>
<BODY>
<%
Set MyContent=Server.CreateObject("IISSample.ContentRotator")
%>
<%=MyContent.ChooseContent("content.txt") %>
</BODY>
</HTML>
```

דוגמה זו נפתחת ביצירת מופע של הרכיב על ידי קריאה לשיטה ()CreateObject. לאחר מכן, מאוחזרת מחרוזת HTML מתוך קובץ תזמון התוכן הנקרא content.txt. מחרוזת זו מוצגת בדף האינטרנט. בפעם הבאה שיידרש דף זה, ייתכן ותוצג מחרוזת שונה.

## הקובץ Content schedule

קובץ תזמון התוכן (Content schedule file) מכיל את כל מחרוזות תוכני HTML. קובץ זה הוא קובץ טקסט רגיל שניתן ליצירה ועריכה על ידי כל עורך טקסט, וניתן לקרוא לו בכל שם המוכר על ידי מערכת ההפעלה. ראה דוגמה בתדפיס 7.4 להלן:

**תדפיס 7.4** הקובץ content.txt

```
%%#2 // Here is the first entry
<FONT COLOR="RED"> Visit Our News Group! </FONT>
%%#3 // Here is the second entry
<B> Don't Forget To Bookmark This Web Site. </B>
%%#5 // Here is the third entry
Download the following free software from our Download Page:
<UL>
<LI> ActiveX Components
<LI> Link Checker
<LI> HTML Validator
</UL>
```

קובץ content schedule זה כולל שלוש רשומות (Entries). תחילתה של כל רשומה מסומנת בשני סימני אחוז (%%). בכל קריאה לשיטה ()ChooseContent מאוחזרת אחת מרשומות אלו.

בדוגמה זו, ניתן לכל ערך משקל מסוים. משקל זה קובע את התדירות היחסית בה תיבחר הרשומה על ידי השיטה ()ChooseContent. המשקל מצוין על ידי ערך מספרי המופיע אחרי סולמית (#). הרשומה הראשונה בקובץ content.txt, למשל, היא בעלת משקל 2.

משקלה של רשומה יכול לנוע בין 0 ל 65,535. רשומה בעלת משקל 0 לא תוצג לעולם (הדבר שימושי אם רוצים לבטל רשומות מסוימות באופן זמני), ככל שיגדל משקל הרשומה כך יגדלו גם סיכוייה להיבחר על ידי השיטה ()ChooseContent. רשומות שלא צוין עבורן משקל הן בעלות משקל של 1.

בדוגמה שלנו, הרשומה הראשונה תוצג פעמיים מתוך כל 10 קריאות לשיטה ()ChooseContent, הרשומה השנייה תוצג 3 פעמים מכל 10, והרשומה השלישית תוצג 5 פעמים מכל 10. כדי לקבוע את התדירות בה תוצג רשומה מסוימת, חלק את משקלה של הרשומה בסכום משקלי כל הרשומות.

בדוגמה, כל רשומה מכילה גם הערה. לדוגמה, הרשומה הראשונה מכילה את ההערה "This is the first entry". כדי לכלול הערה, השתמש בתווים // לפניה. ההערות לא יוצגו בדף האינטרנט.

ולבסוף - כל רשומה מכילה גם מחרוזת מידע HTML. מחרוזת זו יכולה להשתרע על פני מספר שורות, ולכלול את כל תגיות HTML המוכרות. לדוגמה, הרשומה הראשונה תציג בצבע אדום את המחרוזת "Visit Our News Group!", והרשומה האחרונה מכילה רשימת תוכנות הניתנות להורדה מהאתר.

אין חובה לכלול הערות או משקלים. קובץ תזמון תוכן מינימלי יכלול רק מחרוזות HTML, מופרדות על ידי התווים %%. במקרה כזה, כל מחרוזת בקובץ תוצג בתדירות זהה.

# הצגת התוכן של קובץ תזמון התוכן

כדי להציג את כל התוכן (Dumping) של קובץ תזמון התוכן (Content schedule file), אנו צריכים לבצע פעולות מסוימות. הרכיב Content Rotator כולל שיטה נוספת. על ידי שימוש בשיטה ()GetAllContent ניתן לאחזר את **כל** מחרוזות HTML שבקובץ תזמון התוכן. להלן דוגמה לשימוש בשיטה זו:

```
<HTML>
<HEAD><TITLE> Content Schedule File Contents </TITLE></HEAD>
<BODY>
<%
Set MyContent=Server.CreateObject("IISSample.ContentRotator")
%>
<%=MyContent.GetAllContent("content.txt") %>
</BODY>
</HTML>
```

כאשר דף ASP זה מוצג, נכללות בו כל הרשומות הכלולות בקובץ תזמון התוכן content.txt. הרשומות מופרדות באופן אוטומטי על ידי קווים אופקיים הנוצרים על ידי התגית <HR>.

מדוע שתרצה לעשות זאת? שיטה זו יכולה להתגלות כשימושית במקרים מסוימים. למשל, אם אתה משתמש ברכיב להצגת קישורים אקראיים לאתרים, ייתכן ותרצה לספק למשתמש את האפשרות לצפות ברשימת האתרים המלאה.

השיטה ()GetAllContent שימושית גם לניפוי שגיאות (Debugging) בקבצי תזמון תוכן. אם תרצה לבדוק כיצד תיראה כל אחת מהרשומות שבקובץ, תוכל להשתמש בשיטה זו על מנת לצפות בכל הרשומות לפני שאתה משחרר לעולם.

# הרכיב Content Linking

הרכיב **Content Linking** (קישור התוכן) שימושי במצבים בהם יש לך סדרת דפים שעליך לקשר אותם אחד לשני. לדוגמה, תוכל להשתמש ברכיב זה כדי לקשר בין דפים של ספר וירטואלי, מצגת שקופיות, או אפילו הודעות בקבוצות דיון.

**הערה:**

הרכיב Content Linking נכלל בחבילת האופציות של Windows NT. הוא אמור להיות מותקן באופן אוטומטי בעת התקנת Internet Information Server. שמו של הרכיב הוא NEXTLINK.dll, ובברירת המחדל שלו הוא נמצא תחת התיקיה Winnt/system32/inetsrv.

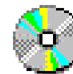
באופן רגיל, כשרוצים לקשר סדרת דפים אחד לשני יש לכלול היפר-קישור בכל אחד מהם. רכיב ה-Content Linking נועד לפשט תהליך זה. על ידי שימוש ברכיב זה ניתן ליצור רשימת דפים בקובץ בודד, ולאחר מכן להשתמש בשיטות של הרכיב להצגת הקישורים המתאימים בכל דף.

הרכיב Content Linking כולל את השיטות הבאות:

- (GetListCount(Content Linking List File - שיטה זו מחזירה את מספר הדפים הכולל בקובץ רשימת קישור התוכן (Content Linking List file).
- (GetListIndex(Content Linking List File - שיטה זו מחזירה את מיקומו של הדף הנוכחי בקובץ רשימת קישור התוכן.
- (GetNextDescription(Content Linking List File - שיטה זו מחזירה את תיאורו של הדף הבא בקובץ רשימת קישור התוכן.
- (GetNextURL(Content Linking List File - שיטה זו מחזירה את כתובתו של הדף הבא בקובץ רשימת קישור התוכן.

- (GetNthDescription(Content Linking List File, Number - שיטה זו מחזירה את תיאורו של דף הנמצא במיקום מסוים בקובץ רשימת קישור התוכן.
- (GetNthURL(Content Linking List File, Number - שיטה זו מחזירה את כתובתו של דף הנמצא במיקום מסוים בקובץ רשימת קישור התוכן.
- (GetPreviousDescription(Content Linking List File - שיטה זו מחזירה את תיאורו של הדף הקודם בקובץ רשימת קישור התוכן.
- (GetPreviousURL(Content Linking List File - שיטה זו מחזירה את כתובתו של הדף הקודם בקובץ רשימת קישור התוכן.

לדוגמה, נניח וברצונך לבנות מדריך בשיטת שלב-אחר-שלב לבישול פסטה. אתה רוצה להקדיש דף ASP נפרד לכל שלב, ולהציג את הדפים לפי הסדר. הרכיב Content Linking יקל עליך את המשימה.



קודם כל, עליך ליצור **קובץ רשימת קישור תוכן**. זהו קובץ טקסט רגיל שביכולתך ליצור ולערוך באמצעות כל עורך טקסט. קובץ זה יכיל את רשימת הדפים שברצונך לקשר. ראה להלן דוגמה כמוצג בתדפיס 7.5. (תמצא את הקובץ pasta.txt בתקליטור המצורף לספר זה, בתיקיה Books\59303\Chapter07).

**תדפיס 7.5** הוראות לבישול פסטה

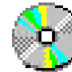
| | |
|---|---|
| /pasta/grabpot.asp | Grab a pot from the cupboard. |
| /pasta/boilwater.asp | Boil some water in the pot. |
| /pasta/openbox.asp | Open box of pasta. |
| /pasta/dumpcontents.asp | Dump contents of box in pot. |
| /pasta/wait.asp | Wait ten minutes. |
| /pasta/home.asp | Return to home page. |

לאחר יצירת קובץ רשימת קישור התוכן, תוכל לשמור אותו בכל שם שתרצה. בדוגמה זו, הקובץ נשמר תחת השם pasta.txt.

קובץ הדוגמה מורכב משני טורים. הטור הראשון מכיל את רשימת הדפים לקישור. אלה יכולים להיות דפי ASP או דפי HTML רגילים. הטור השני מכיל את תיאורי הדפים האלה. על הטורים להיות מופרדים באמצעות תו Tab בודד ולא באמצעות רווחים, כדי לאפשר לרכיב Content Linking להבדיל ביניהם.

**הערה:**

קובץ רשימת קישור התוכן הוא קובץ טקסט רגיל ותו לא. לכן, הרכיב יתעלם מכל עיצוב מיוחד דוגמת כתב מודגש או נטוי שייעשה בו שימוש בקובץ.

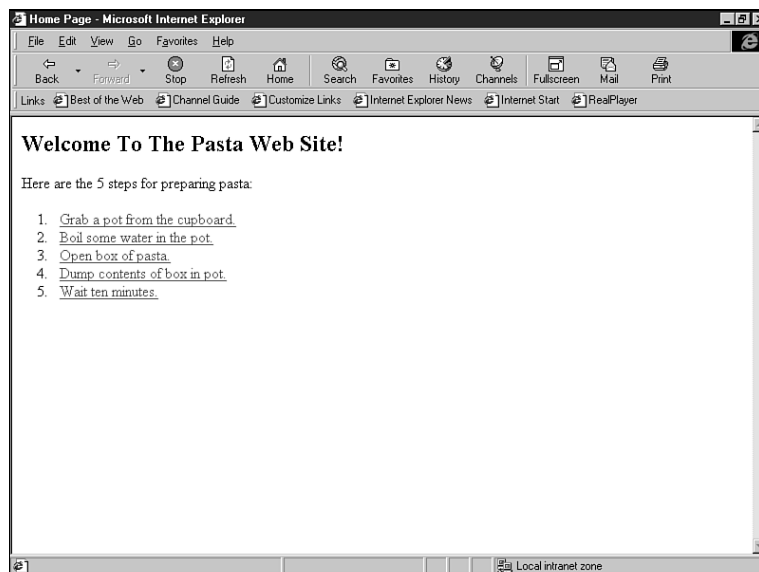


לאחר יצירת קובץ רשימת קישור התוכן, תוכל להשתמש ברכיב Content Linking כדי להוסיף קישורי ניווט לדפי ASP שלך. אם תרצה להציג את רשימת השלבים הנדרשים להכנת פסטה בעמוד הבית של האתר שלך, למשל, תוכל להשתמש בדף ASP המוצג לדוגמה בתדפיס 7.6. (תמצא את הקובץ pastaMenu.asp בתקליטור המצורף לספר זה, בתיקיה Books\59303\Chap07).

**תדפיס 7.6** תפריט לבישול פסטה

```
<HTML>
<HEAD><TITLE> Home Page </TITLE></HEAD>
<BODY>
<H2>Welcome To The Pasta Web Site!</H2>
<%
Set mylinks=Server.CreateObject("MSWC.NextLink")
%>
Here are the
<%=mylinks.GetListCount("pasta.txt")-1%>
steps for preparing pasta:
<OL>
<%
FOR i=1 TO mylinks.GetListCount("pasta.txt")-1
%>
<LI><A HREF="<%=mylinks.GetNthURL("pasta.txt",i)%>">
<%=mylinks.GetNthDescription("pasta.txt",i)%></A>
<%
NEXT
%>
</OL>
</BODY>
</HTML>
```

דף ASP זה מציג את רשימת הקישורים שבקובץ רשימת קישור התוכן (ראה תרשים 7.4). הדבר מתבצע על ידי יצירת מופע של הרכיב Content Linking. מופע הרכיב מוקצה למשתנה בשם mylinks.




**תרשים 7.4** אתר הפסטה

שלוש משיטות הרכיב באות לידי שימוש בדוגמה זו:

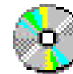
- תחילה, מאחזרת השיטה ()GetListCount את מספרן של הרשומות שבקובץ רשימת קישור התוכן. בכל קריאה לשיטה משיטות הרכיב Content Link יש לציין את שם קובץ רשימת קישור התוכן. בדוגמה זו, הקריאה לשיטה נעשית בשורת הקוד ("mylinks.GetListCount("pasta.txt.

## הערה:

הלולאה FOR האחרונה נעה מ 1 עד ל GetListCount()-1, כך שהיא משמיטה את האיבר האחרון ברשימה. הדבר נעשה באופן מכוון, כיוון שהאיבר האחרון ברשימה הוא קישור שנועד לחזרה לדף הבית.

- לאחר מכן, מאחזרת השיטה ()GetNthURL את רשומת קישור URL ה-Nth שבקובץ רשימת קישור התוכן. שיטה זו מקבלת שני פרמטרים. הפרמטר הראשון מציין את שם קובץ רשימת קישור התוכן, והפרמטר השני מציין איזה איבר לאחזר מהרשימה. לדוגמה, על ידי שימוש ב-GetNthURL("pasta.txt",2), תוכל לאחזר את האיבר השני ברשימה.
- לבסוף, נעשה שימוש בשיטה ()GetNthDescription כדי לאחזר את תיאורי הקישורים מתוך קובץ רשימת קישור התוכן. גם שיטה זו מקבלת שני פרמטרים. הפרמטר הראשון מציין את שמו של קובץ רשימת קישור התוכן. הקובץ השני מציין איזה איבר יש לאחזר מן הקובץ. לדוגמה, על ידי שימוש ב-GetNthDescription("pasta.txt",2), תוכל לאחזר את תיאור האיבר השני שבקובץ pasta.txt.

כדי להציג את כל האיברים שבקובץ רשימת קישור התוכן נעשה שימוש בשיטות GetNthURL() ו- GetNthDescription() בתוך לולאת FOR...NEXT. כל האיברים מלבד האחרון מוצגים. בחרנו שלא להציג את האיבר האחרון, מפני שהוא מצביע בחזרה לדף הבית.



בדוגמה הקודמת, השתמשנו בשיטות של הרכיב Content Linking כדי ליצור רשימה של דפים. ניתן להשתמש בשיטות הרכיב גם כדי לקשר את הדפים זה לזה, כמו שמודגם בתדפיס 7.7. (תמצא את הקובץ pastaStep2.asp בתקליטור המצורף לספר זה, בתיקיה Books\59303\Chap07).

**תדפיס 7.7** השלב השני בהכנת פסטה

```
<HTML>
<HEAD><TITLE> Step Two </TITLE></HEAD>
<BODY>
<H1>Step 2: Boil Water </H1>
<H3>Boil some water in a pot.</H3>
<HR>
<%
Set mylinks=Server.CreateObject("MSWC.NextLink")
IF mylinks.GetListIndex("pasta.txt")>1 THEN
%>
<A HREF="<%=mylinks.GetPreviousURL("pasta.txt")%>">
Previous Step</A>
<% END IF %>
<P>
<A HREF="<%=mylinks.GetNextURL("pasta.txt")%>">
Next Step</A>
</BODY>
</HTML>
```

בדף ASP זה נעשה שימוש בשתי שיטות של הרכיב. השיטה GetPreviousURL() מאחזרת את מיקומו של הדף הקודם, והשיטה GetNextURL() מאחזרת את מיקומו של הדף הבא. שיטות אלו יוצרות קישור לדף הקודם ולדף הבא. הדף הנוכחי קובע את התוצאות שיוחזרו על ידי השיטות GetPreviousURL() ו- GetNextURL(). כשמתבצעת קריאה לאחת משיטות אלו, נתיב הדף הנוכחי מבצע השוואה עם הרשומות בקובץ רשימת קישור התוכן. השיטה GetPreviousURL() מחזירה את הרשומה שמעל רשומת הדף הנוכחי ברשימה, והשיטה GetNextURL() מחזירה את הרשומה שמתחת לרשומת הדף הנוכחי.

במקרה שהדף הנוכחי אינו נכלל בקובץ Content Linking List, תחזיר השיטה GetPreviousURL() את הרשומה הראשונה בקובץ. השיטה GetNextURL() תחזיר את הרשומה האחרונה.

הרכיב Content Linking לא מבצע כל פעולה שלא יכולת לבצע בעצמך על ידי הוספה ידנית של קישורים בכל דף. הוא רק עוזר בתחזוקת הקישורים בין הדפים, בכך שהוא מבצע את הפעולות באופן אוטומטי.

# סיכום

בפרק זה, למדת את אופן השימוש בשלושה רכיבים המשמשים ליצירת תוכן דינמי בדפי ASP. בחלק הראשון, למדת להשתמש ברכיב Browser Capabilities כדי לזהות תכונות של דפדפנים, וגילית מספר ממגבלותיו. בחלק השני, למדת כיצד להציג דפי HTML שונים בתדירויות שונות על ידי שימוש ברכיב Content Rotator. ולבסוף, למדת כיצד להפוך את הניווט באתר שלך לפשוט יותר, על ידי שימוש ברכיב Content Linking.

## פרק 8

# עבודה עם מערכת הקבצים

**בפרק זה:**

- **סקירה כללית של הרכיב File Access**
- **קריאה וכתיבה לקובץ**
- **עבודה עם קבצים**
- **עבודה עם כוננים ותיקיות**

פרק זה מוקדש כולו לרכיבים שבעזרתם ניתן לגשת לקבצים באמצעות ASP. על ידי שימוש ברכיב זה בדפי ASP שלך, תוכל להשיג שליטה מלאה במערכת הקבצים במחשב. חלקו הראשון של הפרק יציג סקירה כללית של הרכיבים בהם משתמשים באובייקט זה. בחלק השני תלמד כיצד לקרוא ולכתוב לקובץ טקסט. בחלק השלישי תלמד לעבוד עם שיטות, מאפיינים ואוספים של קבצים. החלק האחרון של הפרק בוחן שיטות לניהול כוננים ותיקיות.

# סקירה כללית של הרכיב File Access

דפי ASP כוללים אוסף עשיר של **שיטות** (methods), **מאפיינים** (properties) **ואוספים** (collections) לעבודה עם קבצים. על ידי שימוש בתסריטי ASP, תקבל שליטה מלאה במרבית ההיבטים של מערכת הקבצים.

כדי לעבוד עם קבצים, יש להשתמש ברכיב File Access ובאובייקטים הבאים :

- FileSystemObject - כולל את כל השיטות הבסיסיות לעבודה עם מערכת הקבצים. לדוגמה, תוכל להשתמש בשיטות של אובייקט זה כדי להעתיק או למחוק קבצים ותיקיות.
- TextStream - משמש לקריאה וכתיבה של קבצי טקסט.
- File - השיטות והמאפיינים של אובייקט זה מאפשרות לעבוד עם קבצים בודדים. לדוגמה, תוכל להשתמש באובייקט זה כדי למצוא את תאריך השינוי האחרון של קובץ או את נתיבו (Path).
- Folder - השיטות והמאפיינים של רכיב זה מאפשרות לעבוד עם תיקיות.
- Drive - אובייקט זה מייצג כונן במחשב המקומי, או שיתוף רשת. תוכל להשתמש בשיטות ובמאפיינים של אובייקט זה כדי לאחזר מידע, כמו כמות המקום הפנוי בכונן, או סוג מערכת הקבצים שלו.

בנוסף, הרכיב מכיל שלושה אוספים (collections) :

- האוסף Files - מייצג ערכה (Set) של קבצים בתיקיה.
- האוסף Folders - הפריטים שבאוסף זה מייצגים תיקיות משנה בתוך תיקיה.
- האוסף Drives - אוסף זה מייצג את הכוננים המוגדרים בשרת, או כוננים שיתופיים מרוחקים ברשת.

רכיב File Access כולל כמאה מאפיינים ושיטות (הרבה מעבר למה שניתן לכסות בפרק יחיד). לאחר קריאת העמודים הבאים, תבין כיצד להשתמש במאפיינים ובשיטות החשובים ביותר של האובייקטים ברכיב זה, בעת כתיבת תסריטי ASP שלך. רשימה מלאה של השיטות (Methods), המאפיינים (Properties) והאוספים (Collections) באובייקטים אלה תוכל לראות בנספח B, שבסוף ספר זה.

# קריאה וכתיבה לקובץ

בסעיף זה תמצא את תיאור הפעולות החשובות ביותר בגישה לקבצים - קריאה וכתיבה לקובץ טקסט. לקבצי טקסט שימושים רבים, וביניהם :

- יצירת יומן אירועים (Log) מותאם אישית - קובץ log נמצא ומנוהל בדרך כלל על ידי שרתים כמו IIS, והוא נועד למעקב, למשל על כניסות מבקרים לאתר שלך. תוכל לאגור מידע מגוון, כמו לדוגמה : כתובת IP של המבקרים, הדפדפנים בהם השתמשו, ומשך זמן השהייה באתר.
- נתוני טפסים - השתמש בקובץ טקסט לאחסון המידע שהתקבל מטופס HTML. לדוגמה, אם המשתמש מכניס את פרטי ההרשמה שלו לטופס HTML, תוכל לשמור מידע זה בקובץ טקסט.
- טיפ יומי - אחסן בקובץ טקסט רשימה של טיפים למשתמשים באתר שלך, ולאחר מכן אחזר והצג אותם באופן אקראי בדף אינטרנט.

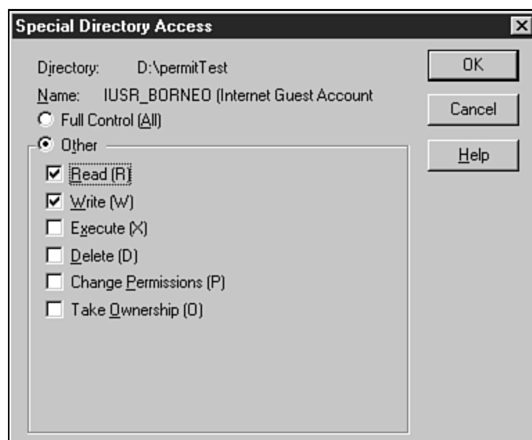
# כתיבה לקובץ טקסט

לפני שתוכל ליצור קובץ בתיקיה, תצטרך הרשאת כתיבה לתיקיה זו. משתמשי אינטרנט אנונימיים ניגשים לאתר שלך על פי **הקשר האבטחה** (security context) של החשבון IUSER_MachineName. כדי לבדוק את הרשאות הכתיבה לתיקיה של חשבון זה, לחץ לחיצה ימנית על שם התיקיה, בחר **Properties** (מאפיינים), לאחר מכן בחר **Security** (אבטחה). לחץ על לחצן **Permissions** (הרשאות), ובדוק את הרשאות החשבון IUSER_MachineName. לחשבון IUSER_MachineName, או לקבוצה המכילה אותו, חייבת להיות הרשאת כתיבה (Write Access) כדי שניתן יהיה ליצור קובץ חדש (ראה תרשים 8.1).

**הערה:**

אם בעת הרצת התסריט מתקבלת הודעת השגיאה הבאה, פירוש הדבר שלחשבון IUSER_MachineName אין הרשאות הדרושות לכתיבת הקובץ.

Writing file...
Server object error 'ASP 0177 : 800a004c'
Server.CreateObject Failed
/mydir/test.asp, line 7
The operation completed successfully.




**תרשים 8.1** קביעת הגדרות משתמש

כדי ליצור קובץ ולכתוב אליו תוכל להשתמש באובייקטים FileSystemObject ו-TextStream. ראשית, עליך ליצור מופע של FileSystemObject. לאחר מכן, קרא לשיטה ()CreateTextFile של האובייקט FileSystemObject כדי להחזיר מופע של אובייקט מסוג TextStream. ניתן בעזרת השיטה ()CreateTextFile ליצור קבצי טקסט ולכתוב אליהם אך אין חובה לתת סיומת (txt.) לקבצים שאנו יוצרים באמצעות שיטה זו. לבסוף, השתמש בשיטה ()WriteLine של האובייקט TextStream כדי לכתוב מידע לקובץ. להלן דוגמה:

```
<HTML>
<HEAD><TITLE>Write File</TITLE></HEAD>
<BODY>
Writing file...
<%
Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
Set MyTextFile=MyFileObject.CreateTextFile("c:\mydir\test.txt")
MyTextFile.WriteLine("Hello There!")
MyTextFile.Close
%>
</BODY>
</HTML>
```

דוגמה זו יוצרת קובץ בשם test.txt בנתיב c:\mydir\test.txt. לאחר מכן, נפעיל את השיטה ()WriteLine לכתיבת שורת הטקסט "!Hello There" אל הקובץ. לבסוף, מופע האובייקט TextStream נסגר, כדי למנוע בזבוז של משאבי מערכת. כל אחד משלבים אלה מתואר ביתר פירוט בפסקאות הבאות.

### הערה:

אם תצטרך למפות נתיב וירטואלי של קובץ לנתיב פיסי, תוכל להשתמש בשיטה ()Server.MapPath, כפי שמוצג בשורה הבאה:

```
PhysicalPath=Server.MapPath("/private/test.txt")
```

אם הנתיב הפיסי של התיקיה הווירטואלית "/private" הוא c:\private, קריאה זו ל- ()MapPath תחזיר "c:\private\test.txt".

השיטה ()CreateTextFile משמשת ליצירת קובץ טקסט חדש. בעת קריאה לשיטה זו, מוחזר אובייקט מסוג TextStream. לשיטה זו פרמטר חובה אחד ושני פרמטרי רשות:

- הפרמטר FileSpecifier - פרמטר זה מציין את הנתיב שבו יוצרים את הקובץ. אם אחת התיקיות בנתיב אינה קיימת בדיסק, מוחזרת הודעת השגיאה File Not Found.
- הפרמטר Overwrite - פרמטר זה הינו רשות והוא מציין דריסה, או כתיבה על (Overwrite). כברירת מחדל, הערך שלו הוא TRUE. קריאה לפונקציה ()CreateTextFile דורסת באופן אוטומטי כל קובץ בעל אותו שם שכבר קיים בתיקיה. אם הפרמטר מוצב ל-FALSE, תתקבל הודעת שגיאה אם קיים כבר קובץ בשם זהה.
- הפרמטר Unicode - פרמטר זה הינו רשות והוא מציין את טבלת קוד התווים בה משתמשים ליצירת הקובץ. כברירת מחדל, הערך שלו הוא FALSE, ופירושו הוא שליצירת הקובץ משתמשים בטבלת קוד ASCII. אם הפרמטר הוא TRUE, משתמשים ליצירת הקובץ בטבלת קוד Unicode.

  Unicode הוא תקן בינלאומי לקידוד תווים במערכת ההפעלה. תקן זה מאפשר למערכת ההפעלה לתמוך במספר תווים רב יותר מאשר תקן ASCII אשר מאפשר רק ל- 256 תווים להיתמך. מכיון שתקן ASCII תומך רק ב-256 תווים, מערכת ההפעלה מסוגלת לתמוך רק בשפה אחת נוספת מלבד השפה האנגלית הסטנדרטית. תקן Unicode נתמך על ידי Windows NT ו-Windows 2000 ואילו תקן ASCII נתמך על ידי Windows 98 ו- Windows 95.

לאחר יצירת הקובץ באמצעות השיטה ()CreateTextFile, ניתן לכתוב בו באמצעות האובייקט TextStream. בעת שימוש באובייקט TextStream לכתיבה, יעמדו לרשותך השיטות הבאות:

- Write(string) - שיטה זו כותבת בקובץ מחרוזת תווים.
- WriteLine([string]) - שיטה זו כותבת בקובץ את המחרוזת string ומוסיפה אחריה תו קוד לשורה חדשה. הפרמטר string הוא רשות. אם לא מצוינת מחרוזת, השיטה כותבת לקובץ את תו שורה חדשה בלבד.
- WriteBlankLines(lines) - שיטה זו כותבת לקובץ מספר מסוים של שורות ריקות (באמצעות תווי שורה חדשה).
- ()Close - שיטה זו משמשת לסגירת קובץ TextStream פתוח, ופינוי משאבי המערכת.

לדוגמה, כדי ליצור קובץ טקסט המכיל את הטקסט 'Hello World!' 32 פעמים ברצף, נכתוב את התסריט הזה:

```
<%
Set MyFileObject=Server.CreateObject("Scripting.FileSystem.Object")
Set MyTextFile=MyFileObject.CreateTextFile("c:\mydir\test.txt")
FOR i=1 to 32
	MyTextFile.WriteLine("Hello World!")
NEXT
MyTextFile.Close
%>
```

# קריאה וצירוף לקובץ טקסט

כדי לקרוא מקובץ, עליך ליצור מופע של האובייקט FileSystemObject. לאחר מכן, השתמש בשיטה ()OpenTextFile להחזרת מופע של אובייקט מסוג TextStream. לבסוף, תוכל להשתמש בשיטה ()ReadLine של האובייקט TextStream כדי לקרוא נתונים מן הקובץ. להלן דוגמה:

```
<%
Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
Set MyTextFile=MyFileObject.OpenTextFile("c:\mydir\test.txt")
WHILE NOT MyTextFile.AtEndOfStream
        Response.Write(MyTextFile.ReadLine)
WEND
MyTextFile.Close
%>
```

תסריט זה קורא את כל תוכנו של הקובץ test.txt. תוכן הקובץ מועבר לדפדפן. אם הקובץ לא קיים, מוחזרת הודעת השגיאה File Not Found. (בסעיף הבא, תלמד כיצד לזהות אם קובץ מסוים קיים או לא).

הלולאה WHILE...WEND בדוגמה זו עוברת על תכולת הקובץ, עד שהיא מגיעה לסופו. כל עוד לא הגיעה הלולאה לסוף הקובץ, ערך המאפיין AtEndOfStream הוא FALSE.

המאפיינים הבאים של האובייקט TextStream שימושיים בשעת קריאה מקובץ טקסט:

- AtEndOfLine - מאפיין זה מציין האם הגיעה סופה של שורה מסוימת בקובץ הטקסט. בעת זיהוי תו שורה החדשה, מקבל מאפיין זה את הערך TRUE.
- AtEndOfStream - מאפיין זה מציין האם הגיע סופו של הקובץ. ערכיו יכולים להיות TRUE או FALSE.
- Column - מאפיין זה מציין את מיקומו של התו הנוכחי בשורה. ערכו הוא מספר שלם.
- Line - מאפיין זה מציין את מספר השורה הנוכחי בקובץ. ערכו הוא מספר שלם.

במקום להשתמש בשיטה ()ReadLine לקריאה מקובץ, ניתן להשתמש בשיטה ()Read. שיטה זו מחזירה מספר מסוים של תווים מקובץ טקסט פתוח. הדוגמה הבאה ממחישה את אופן השימוש בשיטה זו:

```
<%
Set MyFileObject=Server.CreateObject("Scripting.FileSystem.Object")
Set MyTextFile=MyFileObject.CreateTextFile("c:\mydir\test.txt")
WHILE NOT MyTextFile.AtEndOfLine
        Response.Write(MyTextFile.Read(1))
WEND
MyTextFile.Close
%>
```

תסריט זה מאחזר את השורה הראשונה מתוך קובץ טקסט, תו אחר תו. המאפיין AtEndOfLine מזהה את ההגעה לסוף שורת הטקסט הראשונה. השיטה ()Read קוראת תו אחר תו מתוך קובץ הטקסט.

השיטות הבאות שימושיות בעת קריאת מידע מקובץ טקסט:

- Read(characters) - שיטה זו קוראת מספר מסוים של תווים מתוך קובץ טקסט.
- ()ReadLine - שיטה זו קוראת שורה בודדת מתוך קובץ טקסט (תו השורה החדשה אינו מאוחזר).
- ()ReadAll - שיטה זו מאחזרת את כל תכולתו של קובץ TextStream.
- Skip(characters) - שיטה זו מדלגת על המספר המצוין של תווים בקובץ טקסט פתוח.
- ()SkipLine - שיטה זו מדלגת על שורת טקסט בודדת בקובץ טקסט פתוח.
- ()Close - שיטה זו משמשת לסגירת קובץ פתוח מסוג TextStream ופינוי משאבי המערכת.

באופן רגיל, השיטה ()OpenTextFile משמשת לאחזור מידע מתוך קובץ טקסט. אבל, ניתן להשתמש בשיטה זו גם למטרת הוספת (Append) מידע לקובץ, כך:

```
<%
Set MyFileObject=Server.CreateObject("Scripting.FileSystem.Object")
Set MyTextFile=MyFileObject.CreateTextFile("c:\mydir\browser.log",8, TRUE)
MyTextFile=WriteLine(Request.ServerVariables("HTTP_USER_AGENT"));
MyTextFile.Close
%>
```

תסריט זה יוצר קובץ יומן (Log) של סוגי הדפדפנים שניגשו לאתר האינטרנט. בכל הרצה של התסריט, סוג הדפדפן שביקש את הדף נרשם בקובץ טקסט. מידע זה על הדפדפן מאוחזר באמצעות האוסף ServerVariables.

**הערה:**

למידע נוסף על האוסף ServerVariables, ראה פרק 2.

התסריט מוסיף את סוג הדפדפן לסופו של קובץ טקסט בשם browser.log. אם הקובץ browser.log לא קיים בעת הרצת התסריט, הוא נוצר באופן אוטומטי. הדבר מתבצע באמצעות שימוש בשני פרמטרים של השיטה ()OpenTextFile: הפרמטר IOMode והפרמטר Create.

הרשימה הבאה מתארת את כל הפרמטרים של השיטה ()OpenTextFile:

- FileSpecifier - פרמטר זה מציין את שם ונתיב הקובץ לקריאה או הוספה.
- IOMode - פרמטר זה הינו אופציונלי, ומציין האם הקובץ ייפתח לקריאה, כתיבה או הוספה. ערך ברירת המחדל הוא 1 - קריאה. לפתיחת קובץ לכתיבה, קבע לפרמטר את הערך 2. לפתיחת קובץ להוספה, קבע את הערך 8.
- Create - פרמטר זה הינו אופציונלי, ומציין האם יש ליצור קובץ, במידה ואינו קיים. ערך ברירת המחדל של הפרמטר הוא FALSE.
- Format - פרמטר אופציונלי זה מציין את מבנה (Format) הקובץ. ברירת המחדל היא קובץ המשתמש בתווי ASCII (ASCII Character Set), אך ניתן להשתמש גם בתווי UNICODE, על ידי קביעת ערך הפרמטר 1-, או בברירת המחדל של המערכת, על ידי קביעת ערך הפרמטר 2-.

**הערה:**

לא ניתן להשתמש בקבועים עם השיטה ()OpenTextFile בהוצאה הנוכחית של דפי ASP. לדוגמה, הפרמטר IOMode יקבל את הערך 8, אך לא את הקבוע ForAppending.

# יישום לדוגמה

חלק זה מביא דוגמה ליישום המשתמש בשיטות הקריאה והכתיבה לקבצים. לפני כמה שנים, לפני שהפכה לחברה של ביליוני דולרים, נערכה ב**אמזון** תחרות לכתיבת סיפור מקוון בשיתוף פעולה. חלקו הראשון של הסיפור נכתב על ידי ג'ון אפדייק (John Updike), ובכל יום שלחו מבקרים באתר האינטרנט של Amazon.com הצעות למשפטי המשך לסיפור. מדי יום נבחר משפט אחד מבין כל ההצעות, המנצח זכה בסכום של $1000, והסיפור נכתב בשיתוף פעולה.

תחרות זו הייתה רעיון עצום. היא משכה מבקרים לחזור לאתר של Amazon.com בכל יום, ויצרה כמות אדירה של פרסום (בכמה מימי התחרות שהתקיימה שישה שבועות, נאספו באמזון 10,000 הצעות למשפטי המשך לסיפור - וזה היה בשנת 1997). ייתכן ותרצה להוסיף דבר דומה גם לאתר האינטרנט שלך. תדפיס 8.1 מציג דוגמה פשוטה לדרך בה ניתן לעשות זאת.

**תדפיס 8.1** התסריט ל- STORY.ASP

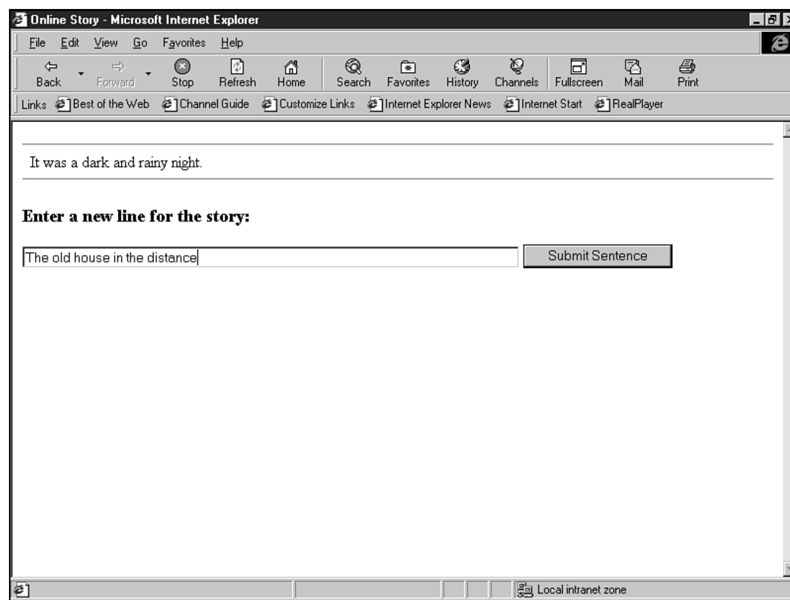
```
<%
IF NOT Request.Form("NextLine")="" THEN
Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
Set MyTextFile=MyFileObject.OpenTextFile("c:\mydir\TheStory.txt", 8, TRUE)
MyTextFile.WriteLine(Request.Form("NextLine"))
MyTextFile.Close
END IF
%>
<HTML>
<HEAD><TITLE> Online Story </TITLE></HEAD>
<BODY>
<HR>
<%
Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
Set MyTextFile=MyFileObject.OpenTextFile("c:\mydir\TheStory.txt")
WHILE NOT MyTextFile.AtEndOfStream
  Response.Write("  "&MyTextFile.ReadLine)
WEND
MyTextFile.Close
%>
<HR>
<H3>Enter a new line for the story: </H3>
<FORM METHOD="POST" ACTION="story.asp">
<INPUT NAME="NextLine" TYPE="TEXT" SIZE=70>
<INPUT TYPE="SUBMIT" VALUE="Submit Sentence">
</FORM>
</BODY>
</HTML>
```

דף ASP זה מורכב משני תסריטים. התסריט הראשון מתבצע בעת שליחת משפט חדש. אם קיים משפט חדש, הוא מתווסף לסוף הקובץ TheStory.txt.

התסריט השני משמש להצגת תכולת הקובץ TheStory.txt. כל שורה משורות הקובץ מועברת לדפדפן (ראה תרשים 8.2). השורות מופרדות באמצעות שני תווי רווח קשיחים (NonBreaking Space characters), כך שהמשפטים יוצגו כשהם מופרדים על ידי רווחים.

שאר דף ASP מכיל טופס HTML לשליחת השורה הבאה בסיפור. דף ASP שולח את תוכן הטופס לעצמו. כדי שהדבר יעבוד, עליך לקרוא לדף ASP בשם Story.asp.

לפני שתשתמש בדף זה בפעם הראשונה, תצטרך ליצור קובץ טקסט בשם TheStory.txt, ולכתוב בו את משפט הסיפור הראשון. כשתרצה להתחיל את הסיפור מן ההתחלה, פשוט מחק את תכולת הקובץ, וכתוב בו משפט ראשון חדש.




**תרשים 8.2** סיפור מקוון בשיתוף פעולה

# עבודה עם קבצים

חלק זה מכסה את נושא העבודה עם קבצים - כיצד להעתיק, להעביר ולמחוק קבצים; כיצד לזהות האם קובץ מסוים קיים; וכיצד לאחזר תכונות (Attributes) של קובץ.

## העתקה, העברה ומחיקה של קבצים

ישנה יותר מדרך אחת להעתיק, להעביר או למחוק קובץ. בכל אחת מפעולות אלו ניתן להשתמש בשיטות האובייקט FileSystemObject או בשיטות האובייקט File. שיטות האובייקט FileSystemObject גמישות מעט יותר, כיון שאינן מגבילות אותך לעבודה עם קובץ אחד בלבד.

הרשימה הבאה מתארת את שיטות האובייקט FileSystemObject המיועדות לטיפול בקבצים:

- CopyFile source, destination, [overwrite] - שיטה זו מעתיקה קובץ ממיקום אחד לאחר. בפרמטר source ניתן להשתמש בתווי הכללה (WildCards) כדי להעתיק יותר מקובץ אחד בכל פעם. הפרמטר האופציונלי Overwrite מציין האם לדרוס קובץ קיים. הוא יכול לקבל את הערכים TRUE או FALSE.
- MoveFile source, destination - שיטה זו מעבירה קובץ ממיקום אחד אחר. בפרמטר source ניתן להשתמש בתווי הכללה כדי להעביר יותר מקובץ אחד בכל פעם. אם הקובץ כבר קיים במיקום היעד, תתקבל הודעת שגיאה.

- DeleteFile FileSpecifier - שיטה זו מוחקת את הקובץ המצוין. ניתן להשתמש בתווי הכללה למחיקה של יותר מקובץ אחד בכל פעם. אם נעשה שימוש בתווי הכללה ולא נמצאו קבצים למחיקה, תתקבל הודעת שגיאה.

כדי שתוכל להשתמש בשיטות אלו, יהיה עליך ליצור מופע של FileSystemObject. הדוגמה הבאה ממחישה את אופן השימוש בכל אחת מן השיטות:

```
<%
' Create an instance of the FileSystemObject object
Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
' Create a file to manipulate
Set MyFile=MyFileObject.CreateTextFile("c:\test.txt")
MyFile.WriteLine("Hello")
MyFile.Close
' Copy the file
MyFileObject.CopyFile "c:\test.txt", "c:\test2.txt"
' Move the file
MyFileObject.MoveyFile "c:\test2.txt", "c:\test3.txt"
' Delete both files
MyFileObject.DeleteFile "c:\test.txt"
MyFileObject.DeleteFile "c:\test3.txt"
%>
```

במקום להשתמש באובייקט FileSystemObject להעתקה, העברה או מחיקה של קבצים, ניתן להשתמש באובייקט File. שיטות אובייקט זה לטיפול בקבצים הן:

- Copy newcopy, [Overwrite] - שיטה זו יוצרת עותק חדש של הקובץ הנוכחי. אם לפרמטר האופציונלי Overwrite נקבע הערך TRUE, קובץ שכבר קיים יידרס.
- Move newcopy - שיטה זו מעבירה את הקובץ הנוכחי למקום אחר. לאחר הקריאה לשיטה, יתייחס הקובץ הנוכחי לקובץ זה.
- Delete - שיטה זו מוחקת את הקובץ הנוכחי.

כדי שתוכל להשתמש בשיטות אלו, יהיה עליך ליצור מופע של האובייקט File. כדי ליצור מופע של האובייקט File, יש להשתמש בשיטה ()GetFile של האובייקט FileSystemObject. לפניך התסריט הקודם, משוכתב כך שישתמש בשיטות האובייקט File:

```
<%
' Create an instance of the FileSystemObject object
Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
' Create a file to manipulate
Set MyFile=MyFileObject.CreateTextFile("c:\test.txt")
MyFile.WriteLine("Hello")
MyFile.Close
' Create an instance of the file object
Set afile=MyFileObject.GetFile("c:\test.txt")
' Copy the file
afile.Copy  "c:\test2.txt"
```

```
'  Move the file
afile.Move  "c:\test3.txt"
'  Delete the original file
afile.Delete
%>
```

# זיהוי האם קובץ קיים

כדי לזהות אם קובץ קיים, ניתן להשתמש בשיטה ()FileExists של האובייקט FileSystemObject. עליך רק להעביר לשיטה את הנתיב הפיסי של הקובץ, והיא תחזיר לך TRUE או FALSE. לפניך דוגמה לאופן השימוש בשיטה זו:

```
<HTML>
<HEAD><TITLE> FileExists Example</TITLE></HEAD>
<BODY>
<%
MySelf=Request.ServerVariables("PATH_TRANSLATED")
' Create instance of the FileSystemObject
Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
If MyFileObject.FieExists(MySelf) THEN
  Response.Write("I exist!")
ELSE
  Response.Write("I do not exist.")
END IF
%>
</BODY>
</HTML>
```

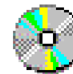
דף ASP זה בודק האם הוא קיים. משתנה השרת PATH_TRANSLATED משמש להחזרת נתיבו הפיסי של הקובץ הנוכחי. השיטה ()FileExists בודקת האם קובץ זה קיים. בדוגמה זו, השיטה תמיד תחזיר TRUE (במילים אחרות, הקובץ בודק את עובדת קיומו, מגלה שהוא קיים, ומחזיר TRUE).

# קבלת תכונות של קבצים

האובייקט File מכיל מספר מאפיינים שמתגלים כשימושיים בעבודה עם קבצים. הרשימה הבאה מסבירה מאפיינים אלה:

- Attributes - מאפיין זה מחזיר את תכונות (Attributes) הקובץ הנוכחי (בדומה לפקודה ATTRIB של DOS). לדוגמה, תוכל להשתמש במאפיין זה כדי לקבוע האם הקובץ הוא נסתר (Hidden) או לקריאה-בלבד (Read-Only).
- DateCreated - מאפיין זה מחזיר את תאריך ושעת יצירת הקובץ.
- DateLastAccessed - מאפיין זה מחזיר את התאריך והשעה בהם ניגשו לקובץ בפעם האחרונה.

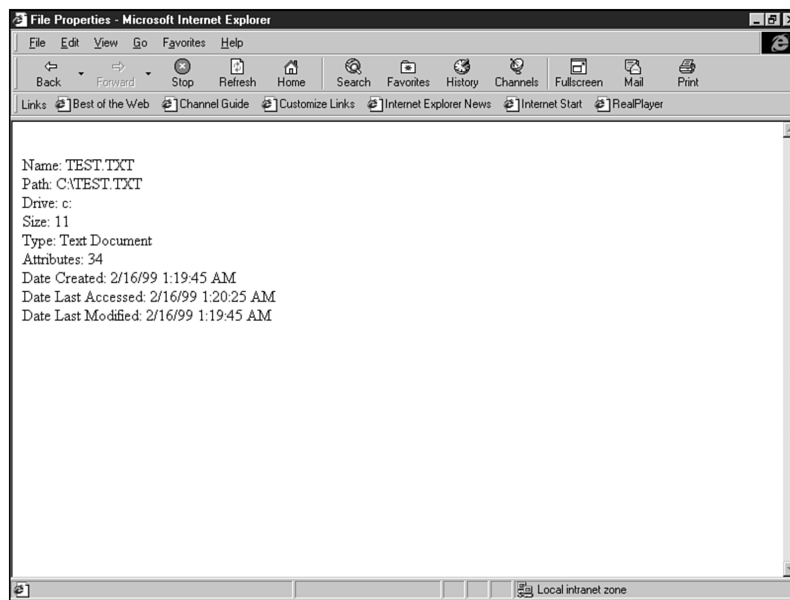
- DateLastModified - מאפיין זה מחזיר את התאריך והשעה בהם שונה קובץ זה בפעם האחרונה.
- Drive - מאפיין זה מחזיר את הכונן בו הקובץ ממוקם.
- Name - מאפיין זה מחזיר את שם הקובץ.
- ParentFolder - מאפיין זה מחזיר את התיקיה בה נמצא הקובץ.
- Path - מאפיין זה מחזיר את נתיבו של הקובץ.
- Size - מאפיין זה מחזיר את גודלו של הקובץ בביטים.
- Type - מאפיין זה מחזיר את סוג הקובץ. לדוגמה, מסמך טקסט (Text Document), קובץ ASP (ASP File) או דף אינטרנט (קובץ HTML).



כדי להשתמש בכל אחד ממאפיינים אלה, יש ליצור מופע של האובייקט File. הדוגמה הבאה מציגה את מאפייני הקובץ שנתיבו c:\test.txt (ראה תרשים 8.3). (תמצא את הקובץ fileProperties.asp בתקליטור המצורף לספר זה, בתיקיה Books\59303\Chap08).

```
<HTML>
<HEAD><TITLE>File Properties</TITLE></HEAD>
<BODY>
<%
'  Create an instance of the FileSystemObject object
Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
'  Create an instance of the File object.
Set afile=MyFileObject.GetFile("c:\test.txt")
%>
<BR>Name: <%=afile.Name%>
<BR>Path: <%=afile.Path%>
<BR>Drive: <%=afile.Drive%>
<BR>Size: <%=afile.Size%>
<BR>Type: <%=afile.Type%>
<BR>Attributes: <%=afile.Attributes%>
<BR>Date Created: <%=afile.DateCreated %>
<BR>Date Last Accessed: <%=afile.DateLastAccessed%>
<BR>Date Last Modified: <%=afile.DateLastModified%>
</BODY>
</HTML>
```

המאפיין Attributes דורש הסבר מסוים. מאפיין זה מחזיר מספר המתאים לסכום התכונות של הקובץ. טבלה 8.1 מפרטת את ערכי התכונות.




**תרשים 8.3** מאפייני קובץ

**טבלה 8.1** תכונות קובץ

| Attribute | ערך | תכונה |
|---|---|---|
| Normal | **0** | רגיל |
| Read-Only | **1** | קריאה-בלבד |
| Hidden | **2** | חבוי/מוסתר |
| System | **4** | קובץ מערכת |
| Volume | **8** | Volume |
| Directory | **16** | תיקיה |
| Archive | **32** | ארכיון |
| Alias | **64** | כינוי |
| Compressed | **128** | דחוס |

לקובץ המוצג בתרשים 8.3 נקבעו התכונות Hidden ו- Archive. השילוב בין 2 (לקובץ נסתר) ו- 32 (לקובץ ארכיון) שווה ל- 34 (2+32=34). לא קיימת סכנת אי בהירות לגבי התכונות, כיון שכל שילוב תכונות יפיק מספר ייחודי.

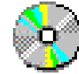
חלק ממאפיינים אלה ניתנים, בנוסף לקריאה, גם לשינוי. ניתן לשנות את המאפיינים לקריאה בלבד, קובץ נסתר וקובץ ארכיון. לדוגמה, כדי להפוך את הקובץ c:\test.txt לקובץ נסתר, תוכל להשתמש בתסריט הבא:

```
<%
'  Create an instance of the FileSystemObject object
Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
'  Create an instance of the File object.
Set afile=MyFileObject.GetFile("c:\test.txt")
'  Make it hidden
afile.attributes=2
%>
```

# עבודה עם כוננים ותיקיות

חלק זה יבחן את השיטות לעבודה עם כוננים ותיקיות. תלמד כיצד לאחזר מידע אודות הכוננים במחשב המקומי, וכיצד ליצור, להעתיק, להעביר ולמחוק תיקיות, ולהציג את תכולתן.

## עבודה עם כוננים

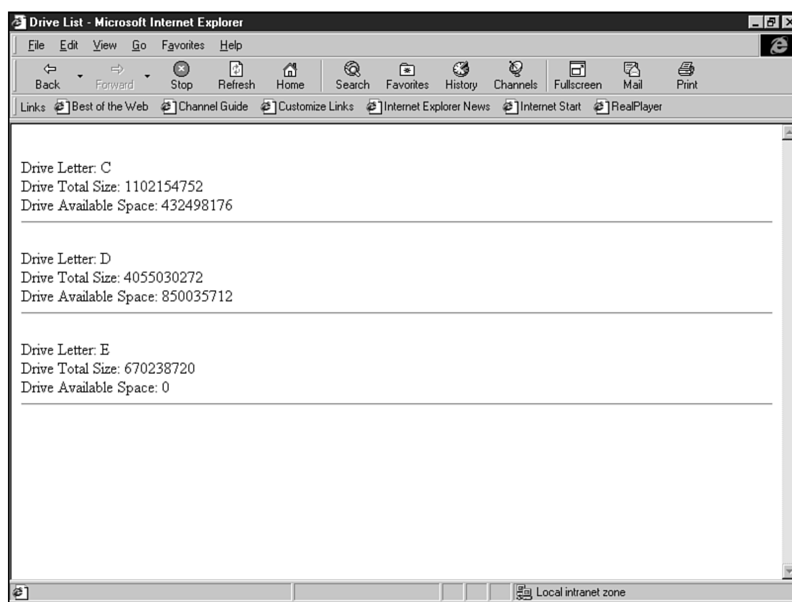


קיימים שני אובייקטים בהם תוכל להשתמש כדי לקבל מידע על כונני המחשב המקומי: האובייקט FileSystemObject והאובייקט Drive. לדוגמה, דף ASP הבא מציג את רשימת כל הכוננים שבשרת, גודלם, והמקום הפנוי בהם. (תמצא את הקובץ driveList.asp בתקליטור המצורף לספר זה, בתיקיה Books\59303\Chap08).

```
<HTML>
<HEAD><TITLE>Drive List</TITLE></HEAD>
<BODY>
<%
'  Create an instance of the FileSystemObject object
Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
'  Loop through the Drives collection
FOR EACH thing in MyFileObject.Drives
%>
<BR>Drive Letter: <%=thing.DriveLetter%>
<BR>Drive Total Size: <%=thing.TotalSize%>
<BR>Drive Available Space: <%=thing.AvailableSpace%>
<HR>
<%
NEXT
%>
</BODY>
</HTML>
```

**הערה:**

אם תריץ תסריט זה כשכונן הדיסקטים או כונן התקליטורים שלך ריק, תקבל הודעת שגיאה. ניתן להתגבר על בעיה זאת על ידי שימוש במאפיין isReady כדי לבדוק האם המדיה נמצאת בכונן.




**תרשים 8.4** הצגת מאפייני כוננים

האוסף Drive באובייקט FileSystemObject מכיל את רשימת הכוננים הזמינים ברשת. אולם, אוסף זה כולל רק את הכוננים שקיבלו מיפוי לאותיות מסוימות. אלו הן שיטות האובייקט FileSystemObject המתייחסות לכוננים:

- DriveExists(DriveSpecifier) - שיטה זו מחזירה TRUE אם הכונן המצוין קיים.
- Drives - שיטה זו מחזירה את אוסף הכוננים במחשב המקומי.
- GetDrive(DriveSpecifier) - שיטה זו מחזירה אובייקט מסוג Drive המייצג את הכונן המצוין.
- GetDriveName(Path) - שיטה זו מחזירה מחרוזת המכילה את הכונן לנתיב המצוין.

באופן לא מפתיע, גם האובייקט Drive מכיל מספר שיטות ומאפיינים השימושיים לעבודה עם כוננים:

- AvailableSpace - מחזיר את המקום הזמין על הכונן בבייטים. (Bytes)
- DriveLetter - מחזיר את אות הכונן. לדוגמה: C:, D: או E:.

- DriveType - מחזיר מספר התואם את סוג הכונן. למשל כונן תקליטורים, או כונן הניתן להסרה.
- FreeSpace - מחזיר את נפח המקום הפנוי בכונן בבתים (בדרך כלל, זהה לערך של AvailableSpace).

**הערה:**

ייתכן ותהית באילו נסיבות יחזירו FreeSpace ו- AvailableSpace תוצאות שונות. מערכת Windows 2000 מאפשרת למנהלי מערכת להקצות לכל אחד ממשתמשי הרשת מכסת נפח אכסון בכונן. אם נפח האכסון המוקצה למשתמש מוגבל, הערכים FreeSpace ו- AvailableSpace יהיו שונים זה מזה.

- IsReady - מציין האם הכונן מוכן לשימוש. מאפיין זה שימושי לציון מצבם של כוננים ברי הסרה.
- Path - מציין את נתיב הכונן.
- RootFolder - מאפיין זה מחזיר אובייקט מסוג Folder, המייצג את תיקיית השורש (Root Folder) של הכונן.
- SerialNumber - מחזיר את מספרו הסידורי של הכונן.
- TotalSize - מחזיר את גודלו הכולל של הכונן בבתים.
- VolumeName - מחזיר מחרוזת המייצגת את שם הכונן.

כדי להשתמש במאפיינים ושיטות אלה, יש ליצור מופע של האובייקט Drive. ניתן לעשות זאת על ידי שימוש בשיטה ()GetDrive של האובייקט FileSystemObject. הדוגמה הבאה מחזירה את שמו של כונן C:

```
<%
' Create an instance of the FileSystemObject object
Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
' Create an instance of the Drive object.
Set MyDrive=MyFileObject.GetDrive("c:")
Response.Write(MyDrive.VolumeName)
%>
```

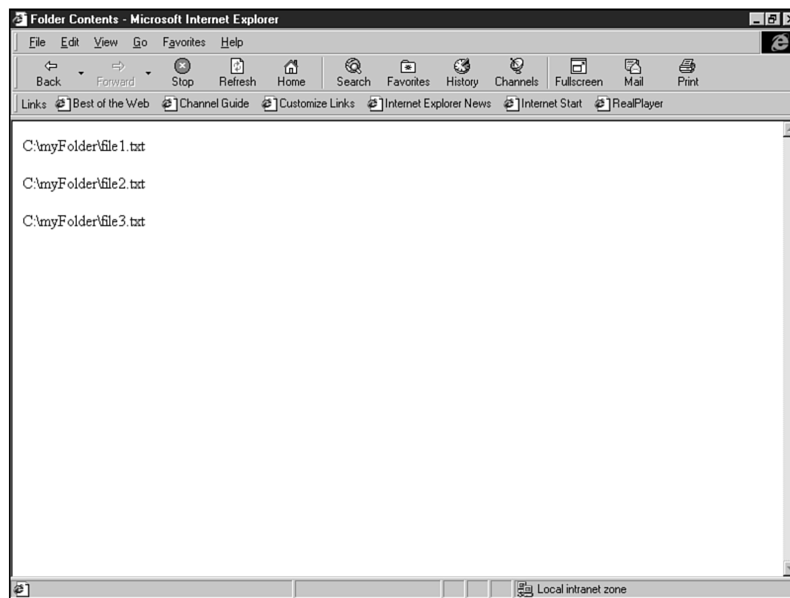
# עבודה עם תיקיות

חלק זה ידגים כיצד לתפעל תיקיות ולהציג את תכולתן. כדי לעבוד עם תיקיות, ניתן להשתמש באובייקט FileSystemObject או באובייקט Folder. דוגמה זו מציגה את כל הקבצים בתיקיה שנתיבה c:\myfolder:

```
<HTML>
<HEAD><TITLE>Folder Contents</TITLE></HEAD>
<BODY>
<%
' Create an instance of the FileSystemObject object
Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
```

```
'  Create a Folder object.
Set MyFolder=MyFileObject.GetFolder("c:\myfolder")
'  loop through the Files collection
FOR EACH thing in MyFolder.Files
  Response.Write("<P>"&thing)
NEXT
%>
</BODY>
</HTML>
```

בדוגמה זו, נוצר אובייקט מסוג Folder על ידי שימוש בשיטה ()GetFolder של האובייקט FileSystemObject. לאחר יצירת האובייקט Folder, משמשת הלולאה FOR...NEXT למעבר על אוסף הקבצים שלו. הדף מציג את כל הקבצים שבאוסף זה (ראה תרשים 8.5).




**תרשים 8.5** תכולת תיקיה

האובייקט FileSystemObject כולל מספר שיטות לעבודה עם תיקיות. הרשימה הבאה מספקת הסבר קצר על דרכי שימוש אפשריות בשיטות אלו:

- CopyFolder source, destination, [overwrite] - שיטה זו מעתיקה תיקיה ממיקום אחד לאחר. ניתן להשתמש בתווי הכללה בפרמטר source, כדי להעתיק מספר תיקיות בו זמנית. כברירת מחדל, תיקיה שכבר קיימת תדרס. כדי למנוע זאת, יש לקבוע את ערך הפרמטר Overwrite כ- FALSE.
- CreateFolder FolderSpecifier - שיטה זו תיצור את התיקיה המצוינת.
- DeleteFolder FolderSpecifier - שיטה זו תמחק את התיקיה המצוינת ואת כל תכולתה. ניתן להשתמש בתווי הכללה כדי למחוק מספר תיקיות בו זמנית.

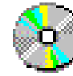
- FolderExists(FolderSpecifier) - שיטה זו מחזירה TRUE אם התיקיה קיימת.
- GetFolder(FolderSpecifier) - שיטה זו מחזירה אובייקט מסוג Folder המייצג את התיקיה המצוינת.
- GetParentFolderName(Path) - שיטה זו מחזירה מחרוזת המכילה את נתיב התיקיה ברמה אחת למעלה.
- MoveFolder source, destination - שיטה זו מעבירה תיקיה ממקום למקום. ניתן להשתמש בתווי הכללה בפרמטר source כדי להעביר יותר מתיקיה אחת בו זמנית.

כדי להשתמש בכל אחת משיטות אלו, יש ליצור תחילה מופע של האובייקט FileSystemObject. הדוגמה הבאה יוצרת תיקיה, מעבירה אותה, ולאחר מכן מוחקת אותה:

```
<%
' Create an instance of the FileSystemObject object
Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
' Create a new folder
MyFileObject.CreateFolder "c:\newfolder"
' Move the folder
MyFileObject.MoveyFolder "c:\newfolder", "c:\oldfolder"
' Delete the folder
MyFileObject.DeleteFolder "c:\oldfolder"
%>
```

גם שיטות ומאפייני האובייקט Folder יכולים לשמש לתפעול תיקיות. לפניך הסבר קצר על המאפיינים והשיטות של האובייקט Folder:

- CopyFolder newcopy, [Overwrite] - שיטה זו מעתיקה את התיקיה הנוכחית למיקום חדש. אם ערך הפרמטר Overwrite נקבע כ-FALSE, תיווצר שגיאה במידה ותיקיית היעד כבר קיימת.
- DeleteFolder - מחיקת את התיקיה הנוכחית.
- Files - מחזיר את אוסף הקבצים בתיקיה. האוסף לא כולל קבצים נסתרים.
- IsRootFolder - מחזיר TRUE אם התיקיה היא התיקיית השורש.
- MoveFolder FolderSpecifier - שיטה זו מעבירה את התיקיה ממיקום אחד לאחר.
- Name - מחזיר את שם התיקיה.
- ParentFolder - מחזיר את תיקיית-האם.
- Size - מחזיר את גודל התיקיה וכל תת-התיקיות שלה בבתים.
- SubFolders - מחזיר את אוסף תת-התיקיות של התיקיה הנוכחית.



כדי להשתמש בכל אחת משיטות אלו, יש ליצור מופע של האובייקט Folder. דוגמה זו מחזירה את רשימת תתי-התיקיות של התיקיה הנמצאת בנתיב c:\myfolder. (תמצא את הקובץ folerContents.asp בתקליטור המצורף לספר זה, בתיקיה Books\59303\Chap08).

```
<HTML>
<HEAD><TITLE>Folder Contents</TITLE></HEAD>
<BODY>
<%
'  Create an instance of the FileSystemObject object
Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
'  Create a folder object
Set MyFolder=MyFileObject.GetFolder("c:\myfolder")
'  Loop through the Files collection
FOR EACH thing in MyFolder.Files
  Response.Write("<P>"&thing)
NEXT
%>
</BODY>
</HTML>
```

הלולאה FOR...EACH בתסריט זה עוברת על כל האלמנטים של האוסף SubFolders, ומציגה את שמם בזה אחר זה.

# סיכום

פרק זה תיאר את השימוש באובייקטים של הרכיב File Access. למדת לקרוא ולכתוב קבצי טקסט, וכיצד לעבוד עם המאפיינים והשיטות של האובייקט File. לאחר מכן, למדת כיצד להשתמש באובייקטים Folder ו- Drive, כדי לתפעל תיקיות ולאחזר מידע לגבי כונני הרשת שלך.

## פרק 9

# מעקב אחר תצוגות הדף, והצגת פרסומות

**בפרק זה:**

- **ספירת מבקרים עם הרכיב Counter**
- **מעקב אחר תצוגות הדף עם הרכיב Counters**
- **הצגת פרסומות**

פרק זה בוחן שלושה רכיבי ActiveX בהם תוכל להשתמש למעקב אחר מבקרים ולהצגת כרזות פרסומת. בחלק הראשון תלמד כיצד להשתמש ברכיב Counter כדי לעקוב אחר מספר הבקשות עבור דף ASP מסוים. לאחר מכן, תלמד להשתמש ברכיב Counters, שיכול לשמש למעקב אחר תצוגות דפים או כל דבר אחר. חלקו האחרון של הפרק מסביר כיצד תוכל להשתמש ברכיב Ad Rotator להצגת כרזות פרסומת באתר האינטרנט שלך.

# ספירת מבקרים באמצעות הרכיב Counter

אם גלשת באקראי ברחבי האינטרנט, ודאי הבחנת כי הדפים הראשיים באתרי אינטרנט רבים כוללים מונה כניסות לאתר. מונה זה מופיע בדרך כלל בתחתית הדף, ומציג את מספר הפעמים בהם ביקרו בדף זה. Microsoft מספקת רכיב המאפשר מימוש פשוט של מונה כזה – הרכיב Page Counter (מונה דפים).

**הערה:**

הרכיב Page Counter אינו נתמך על ידי Microsoft באופן ישיר, ואינו נכלל ב- Option Pack של Windows NT. במשך תקופות מספר בשנה רכיב זה נמצא זמין להורדה מהכתובת http://www.microsoft.com/iis. אם אינך מצליח למצוא אותו באתר של Microsoft, תאלץ לרכוש את Microsoft Internet Information Server Resource Kit.

מכיון שהרכיב Page Counter נועד למטרה מסוימת מאוד, הוא רכיב פשוט. הרכיב כולל את שלוש השיטות הבאות:

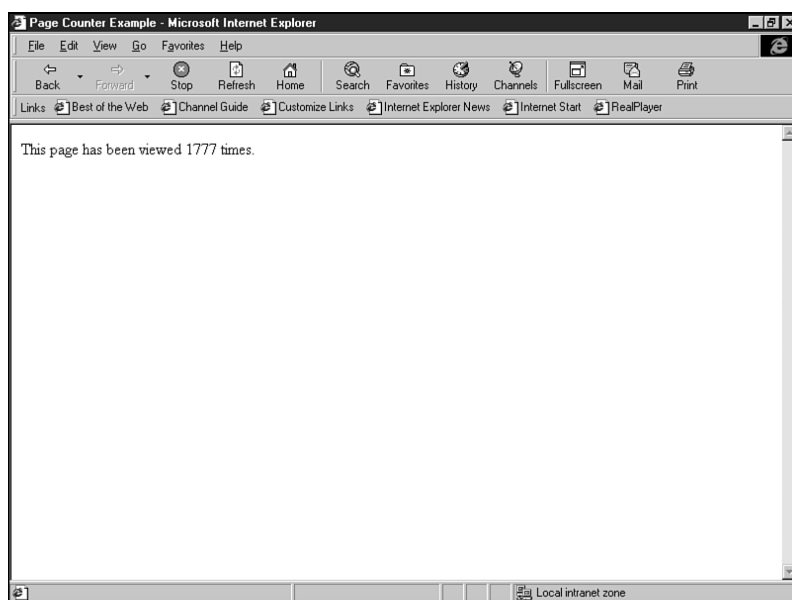
- Hits(Path) - שיטה זו מחזירה את מספר הפעמים בהן נפתח דף בעל נתיב מסוים. אם לא מסופק הנתיב, מוחזר הנתון אודות הדף הנוכחי.
- PageHit() - שיטה זו מעדכנת את מספר הפעמים בהן נפתח הדף הנוכחי. לא ניתן לציין נתיב לדף אחר.
- Reset(Path) - שיטה זו מאפסת את מונה הדף שנתיבו צוין. אם לא מסופק נתיב, תאפס השיטה את מונה הדף הנוכחי.

כיון שהרכיב Page Counter אינו רכיב מובנה דוגמת Request או Response, עליך ליצור מופע שלו לפני שתוכל להשתמש בו. תדפיס 9.1, לדוגמה, יוצר מופע של הרכיב ולאחר מכן מציג את מספר הפעמים בהן הדף נצפה.

תדפיס 9.1 PAGECOUNTER.ASP

```
<HTML>
<HEAD><TITLE> Page Counter Example </TITLE></HEAD>
<BODY>
<%
Set MyHits=Server.CreateObject("IISSample.PageCounter")
MyHits.PageHit()
%>
This page has been viewed
<%=MyHits.Hits%>
times.
</BODY>
</HTML>
```

דף ASP זה פשוט מציג את מספר הפעמים בהן הוא התבקש. השיטה PageHit() משמשת לקידום המונה עבור הדף הנוכחי. השיטה Hit נקראת כאן ללא פרמטרים, כך שהיא מחזירה את ערך המונה עבור הדף הנוכחי (ראה תרשים 9.1).

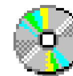



תרשים 9.1 פלט מן הרכיב Page Counter

**הערה:**

משמעות המילה כניסה (Hit) בהקשר של אתרי אינטרנט, מעורפלת למדי ויש להימנע ממנה. מצד אחד, כניסה יכולה להתייחס לכל פעם בה אוחזר רכיב של דף HTML. לדוגמה, ייתכן שדף HTML המכיל 20 תמונות יחולל 20 כניסות (זאת כיוון שהדפדפן יוצר תקשורת עם ה-Server, כלומר שולח בקשת HTTP עבור כל תמונה שאמורה להיות מוצגת בדף). מצד שני, כניסה יכולה להתייחס גם לצפייה בדף. במובן זה, דף HTML המכיל 20 תמונות יחשב כניסה אחת בלבד.

מספר גורמים מנצלים ערפול זה כדי לנפח באופן מלאכותי את הנתונים הסטטיסטיים של האתר שלהם. במקרים קיצוניים, מפתחי אתרים שתלו באתר שלהם תמונות שקופות, אך ורק כדי להגדיל את מספר ה"כניסות" לאתר שלהם. אל תיפול בפח!

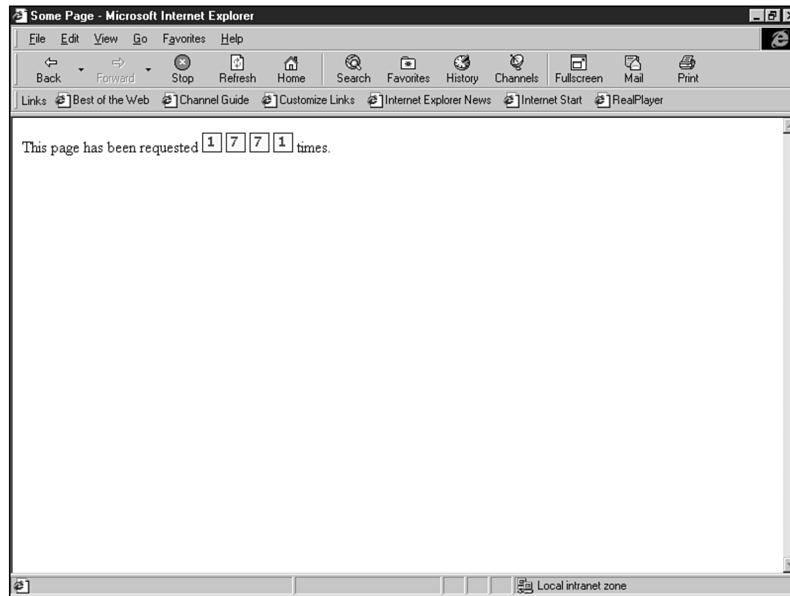


יש להודות כי המונה בתרשים 9.1 הוא מעט משעמם. רוב מוני הכניסות בהם תיתקל באתרי האינטרנט השונים משתמשים בתמונות כדי להציג את מספר הכניסות לאתר. קיימות שתי דרכים לגרום לרכיב Page Counter להציג תמונות. בפרק 16, תלמד ליצור תמונות JPEG תוך כדי ביצוע משימה כלשהי, תוך שימוש ברכיב ShotGraph. לחילופין, תוכל פשוט ליצור סדרה בת 10 תמונות קטנות, אחת לכל סיפרה בין 0 ל 9, ולהשתמש ברכיב Page Counter כדי להציג תמונות אלו - כמתואר בתסריט שבתדפיס 9.2 (ניתן למצוא את הקובץ pagecounterImage.asp בתקליטור המצורף לספר זה).

**תדפיס 9.2** PAGECOUNTERIMAGES.ASP

```
<%
Set MyHits=Server.CreateObject("IISSample.PageCounter")
MyHits.PageHit()
SUB ShowImageCnt(TheNum)
    CntStr=CSTR(TheNum)
    FOR i=1 TO LEN(CntStr)
    CntPart=MID(CntStr,i,1)
    %>
    <IMG SRC="<%=CntPart%>.gif" ALT="<%=CntPart%>">
    <%
    NEXT
END SUB
%>
<HTML>
<HEAD><TITLE>Some Page</TITLE></HEAD>
<BODY>
This page has been requested
<%
ShowImageCnt MyHits.Hits
%>
times.
</BODY>
</HTML>
```

גם דף ASP זה מציג מונה כניסות. אך בדף זה מספר הכניסות מיוצג על ידי תמונות במקום באמצעות טקסט (ראה תרשים 9.2). השגרה ShowImageCnt ממירה את מספר הכניסות למחרוזת, ולאחר מכן נעשה שימוש בלולאה FOR...NEXT כדי להציג את התמונה המתאימה לכל אחת מהספרות שבמחרוזת זו.




**תרשים 9.2** מונה כניסות המציג באמצעות תמונות

כדי להשתמש בדוגמה זו, תצטרך 10 תמונות, הנקראות 0.gif, 1.gif, 2.gif וכן הלאה. תוכל ליצור תמונות אלו בעצמך, או להורידן מרשת האינטרנט. קיימים אתרי אינטרנט רבים המציעים ספריות של תמונות-מונה להורדה חופשית.

# מעקב אחר תצוגות הדפים עם הרכיב Counters

ניתן להשתמש ברכיב Counters (מונים) למניית מספר הבקשות לדף אינטרנט מסוים, אך ניתן גם למנות באמצעותו כל דבר אחר. לדוגמה, תוכל להשתמש בו למניית מספר המבקרים באתר שלך, מספר הפעמים בהם לחצו על פרסומת מסוימת, או אפילו מספר הפעמים בהם נתקבלה בקשה לדף כלשהו על ידי דפדפן כלשהו.

ניתן ליצור מופע אחד בלבד של רכיב זה. אולם, לאחר יצירת המופע, תוכל ליצור כמה מונים שרק תרצה. הרכיב Counters יכול לכלול מספר רב של מונים בעלי שמות שונים.

**הערה:**

הרכיב Counters אינו נתמך על ידי Microsoft באופן ישיר, ואינו נכלל ב- Option Pack של Windows NT. מפעם לפעם, ניתן למצוא רכיב זה להורדה באתר http://www.microsoft.com/iis. אם אינך מצליח למצוא את הרכיב באתר של Microsoft, תאלץ לרכוש את Microsoft Internet Information Server Resource Kit.

היות ובאפשרותך ליצור מופע בודד של הרכיב Counters בלבד, יהיה זה רעיון טוב ליצור את הרכיב בקובץ Global.asa. הדבר יבטיח כי מופע אחד בלבד של הרכיב יווצר בעת הפעלת שרת האינטרנט.

**הערה:**

למידע נוסף אודות השימוש בקובץ Global.asa, פנה לפרק 4.

לפניך דוגמה לאופן יצירת הרכיב בתוך הקובץ Global.asa :

```
<OBJECT RUNAT="Server" SCOPE="Application"  ID="MyCount"
PROGID="MSWC.Counters"></OBJECT>
```

השתמשנו בתגית <OBJECT> המורחבת של Microsoft כדי ליצור מופע בשם MyCount של הרכיב Counters, בטווח היישום כולו. זכור להשתמש בתגית <OBJECT> **מחוץ לתסריטים** שבקובץ Global.asa. לאחר יצירת מופע של הרכיב Counters באופן זה, תוכל לגשת לכל אחת מהשיטות בו מכל דף ביישום, כאילו היה רכיב דפי ASP מובנה.

הרכיב Counters כולל ארבע שיטות. הרשימה הבאה מפרטת את אופן השימוש בכל שיטה :

- Get(counter, name) - שיטה זו מחזירה את ערכו הנוכחי של מונה. אם המונה לא קיים, הוא נוצר וערכו נקבע ל- 0.
- Increment(counter, name) - שיטה זו מוסיפה 1 לערכו הנוכחי של מונה. אם המונה לא קיים, הוא נוצר וערכו נקבע ל- 1.
- Remove(counter, name) - שיטה זו מסירה מונה.
- Setcounter name, integer - שיטה זו מקבלת שני פרמטרים. הראשון הוא שם המונה, והשני הוא ערך מספרי שלם. השיטה מוסיפה את הערך המספרי למונה. אם המונה לא קיים, הוא נוצר ונקבע לערך המצוין.

לאחר יצירת מופע של הרכיב Counters בקובץ Global.asa, תוכל להגדיל ולהקטין מונים שונים בכל אחד מדפי ASP. מונה שנוצר בדף מסוים יכול להיות מוגדל, מוקטן או מושמד מדף אחר. לפניך דוגמה לאופן השימוש ברכיב Counters למעקב אחר מספר הפעמים בהן התבקש דף מסוים:

```
<HTML>
<HEAD><TITLE> Some Page </TITLE></HEAD>
<BODY>
This page has been requested
<%=MyCount.Increment("PageCnt") %>
times.
</BODY>
</HTML>
```

בפעם הראשונה שדף זה מוצג, נוצר מונה בשם PageCnt, וערכו נקבע ל- 1. בקשות חוזרות לדף מגדילות מונה זה ב- 1. המונה PageCnt משקף את מספר הפעמים בהן הדף הוצג.

מה קורה אם השרת קורס במפתיע? המונים שיצרת באמצעות הרכיב Counters שומרים על ערכם. הם שמורים בקובץ בשם counters.txt. אם השרת נופל, המונים עדיין קיימים ושומרים על ערכם, וכשהוא עולה שוב הם ממשיכים מהיכן שהפסיקו.

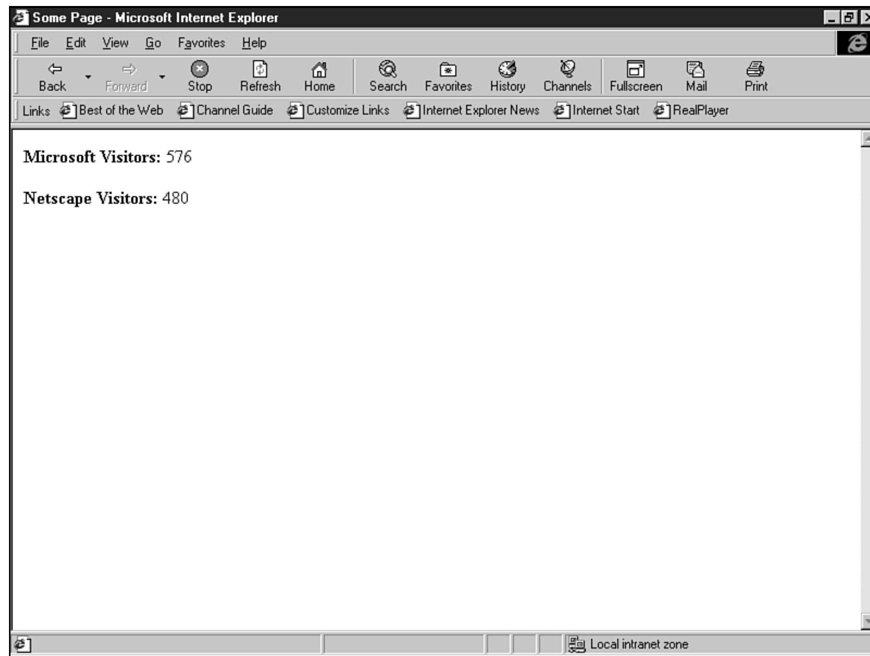
כפי שהוזכר קודם, הרכיב Counters יכול לשמש למעקב אחר כל ערך שתבחר. לדוגמה, תוכל לנהל מעקב אחר מספר הפעמים שמבקר ביקש דף באמצעות דפדפן של Netscape או Microsoft, ולהציג את התוצאות. ראה דוגמה בתדפיס 9.3. (תמצא את הקובץ NsorIE.asp בתקליטור המצורף לספר זה, בתיקיה Books\59303\Chap09).

**תדפיס 9.3** NSorIE.ASP

```
<%
theBrowser = UCASE( Request.ServerVariables( "HTTP_USER_AGENT" ) )

if instr( theBrowser, "MSIE" ) > 0 THEN
  MyCount.Increment( "Microsoft" )
else
  MyCount.Increment( "Netscape" )
end if
%>
<HTML>
<HEAD><TITLE>Some Page</TITLE></HEAD>
<BODY>
<p><b>Microsoft Visitors:</b><%=MyCount.Get( "Microsoft" ) %>
<p><b>Netscape Visitors:</b><%=MyCount.Get( "Netscape" )%>
</BODY>
</HTML>
```

דף ASP זה משתמש בכותרת ה-USER-AGENT לקביעת סוג הדפדפן שביקש את הדף. אם הכותרת כוללת את המחרוזת "MSIE", הדפדפן שביקש את הדף היה Microsoft Internet Explorer. אם לא, ההנחה היא שדפדפן של Netscape ביקש את הדף.

שני מונים הנקראים Microsoft ו- Netscape משמשים לאגירת המידע אודות הפופולריות היחסית של שני הדפדפנים. ערכי המונים מוגדלים על ידי שימוש בשיטה Increment(), ומוצגים על ידי שימוש בשיטה Get() (ראה תרשים 9.3).




**תרשים 9.3** דפים שהתבקשו על ידי הדפדפן של Microsoft לעומת דפים שהתבקשו על ידי הדפדפן של Netscape.

# הצגת פרסומות

אתרי אינטרנט מסחריים רבים בנויים על ההנחה שיוכלו להרוויח כסף על ידי הצגת כרזות פרסומת. ההנחה היא שאתר אינטרנט דומה מאד לעיתון או לתוכנית טלוויזיה. בדומה לתוכנית טלוויזיה, אתר אינטרנט בעל תוכן מעניין ימשוך ציבור צופים. ואחרי הצופים יגיעו גם פרסומות - והכסף של מפרסמים.

אולם, לפי שעה, ברוב אתרי הרשת אסטרטגיה זו לא הוכיחה את עצמה. הבעיה היא שמעט מאוד חברות מוכנות להשקיע בפרסום באינטרנט, ואף גרוע מכך – החברות שמוכנות להשקיע בפרסום מסוג זה, נוטות לפרסם בקבוצה מצומצמת של מספר אתרים בלבד, קבוצה הכוללת את האתר !Yahoo, האתר Netscape, ושערי אינטרנט נוספים דוגמת Lycos, Infoseek, Excite ודומיהן. קבוצה זו מושכת את מרבית כספי המפרסמים, ומשאירה מעט מאוד כספי פרסום לבעלי האתרים הקטנים.

**הערה:**

דרך אחת להתחיל למכור כרזות פרסומת באתר שלך בלי להתאמץ יותר מדי היא להצטרף לרשת פרסום (Advertising Network)
דוגמת Flycast (בכתובת http://www.flycast.com)
או LinkExchange (בכתובת http://www.linkexchange.com).

מעבר לכך, מכירת כרזות פרסומת היא הדרך העיקרית לעשיית כסף באינטרנט. אלא אם אתה מתכנן הכנסה ישירה ממכירת מוצרים או שירותים למבקרי האתר שלך – תוכנית בעלת סיכון רב יחסית, כי תצטרך לשכנע את המבקרים לספק את פרטי כרטיסי האשראי שלהם – תאלץ להשתמש בכרזות פרסומת. בסעיף זה, תלמד כיצד לשלב כרזות פרסומת באתר האינטרנט שלך.

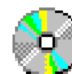
## שימוש ברכיב Ad Rotator

על ידי שימוש ברכיב Ad Rotator (סבב פרסומות), תוכל ליצור דף אינטרנט שיציג פרסומת אחרת בכל פעם שיצפו בו. תוכל לקבוע משקלים שונים לפרסומות שונות, כך שיוצגו בתדירויות שונות. תוכל גם לעקוב אחר מספר הפעמים שפרסומת מסוימת נלחצה כדי לקבוע את קצב הכניסה לפרסומת.

הרכיב Ad Rotator כלול ב-Option Pack של Windows NT. הרכיב אמור להיות מותקן באופן אוטומטי בעת התקנת ה-Internet Information Server (חפש קובץ בשם ADROT.dll בתיקיה Winnt/system32/inetsrv).

**הערה:**

מפרסמים נוהגים למדוד את יעילות הפרסומת באתר אינטרנט לפי שיעור הכניסות (המכונה גם שיעור התעבורה) אליה. שיעור הכניסות (Click-Through Rate) מראה את מידת התכיפות בה משתמש הצופה בפרסומת, מתעניין בה מספיק כדי ללחוץ עליה ולקבל מידע נוסף. שיעור הכניסות הוא היחס בין מספר הפעמים שהפרסומת נלחצה לבין מספר הפעמים שהוצגה.



לרכיב Ad Rotator שיטה אחת בלבד. השיטה ()GetAdvertisement המשמשת לאחזור מידע/נתונים אודות כרזת פרסום. לפניך דוגמה לאופן השימוש בשיטה זו (הדוגמה נמצאת בתקליטור, בקובץ בשם adrot.asp):

```
<HTML>
<HEAD><TITLE> Home Page </TITLE></HEAD>
<BODY>
<CENTER><H1>Welcome to our web site!</H1></CENTER>
<HR>
<%
Set MyAd=Server.CreateObject("MSWC.AdRotator")
%>
<CENTER><%= MyAd.GetAdvertisement("adrot.txt") %></CENTER>
</BODY>
</HTML>
```

דף ASP זה מציג כרזת פרסומת בתחתית הדף (ראה תרשים 9.4). התסריט יוצר מופע של הרכיב Ad Rotator על ידי שימוש בשיטה ()Server.CreateObject. לאחר מכן, מוצגת כרזת הפרסום על ידי קריאה לשיטה ()GetAdvertisement.

שים לב כי השיטה ()GetAdvertisement לוקחת פרמטר. פרמטר זה מציין קובץ המכיל את המידע אודות הפרסומת שתוצג. בדוגמה זו, שם הקובץ הוא adrot.txt. בסעיף הבא תלמד כיצד ליצור קבצים מסוג זה.

# מאפייני הרכיב Ad Rotator

לרכיב Ad Rotator שלושה מאפיינים. ניתן לקבוע את ערכם של מאפיינים אלה לפני הקריאה לשיטה ()GetAdvertisement, כדי לקבוע את האופן בו תוצג הפרסומת. הרשימה הבאה מסבירה כל אחד ממאפיינים אלה:

- Border - מאפיין זה עוקף את הפרמטר BORDER המצוין בקובץ תזמון הסבב (Rotator Schedule File). ניתן להשתמש במאפיין לקביעת גודל הגבול (בפיקסלים) של כרזת הפרסום בדף מסוים.

**תרשים 9.4** דוגמה לכרזת פרסומת

- Clickable - מאפיין זה מציין האם הכרזה תשמש כקישור (Hyperlink). ערך המאפיין יכול להיות TRUE או FALSE.

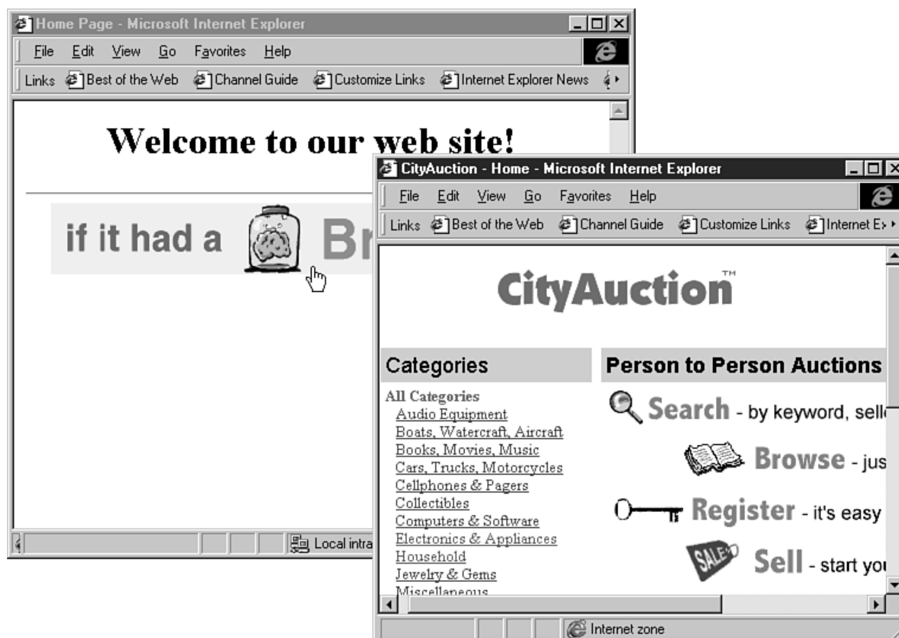
- TargetFrame - מאפיין זה מציין את שם המסגרת אליה ייטען הקישור.

המאפיין TargetFrame שימושי במיוחד. אם תקבע את ערכו של מאפיין זה לשם של מסגרת חדשה, ייטען דף המפרסם, לאחר לחיצה של המשתמש על הפרסומת, למסגרת החדשה (ראה תרשים 9.5). הדבר מונע מהמשתמש לעזוב את האתר שלך לאחר הלחיצה על פרסומת. לפניך דוגמה לאופן השימוש במאפיין זה:

```
<HTML>
<HEAD><TITLE> Home Page </TITLE></HEAD>
<BODY>
<CENTER><H1>Welcome to our web site!</H1></CENTER>
<HR>
<%
Set MyAd=Server.CreateObject("MSWC.AdRotator")
MyAd.TargetFrame= "TARGET=" "AdFrame" " "
%>
<CENTER><%= MyAd.GetAdvertisement("adrot.txt") %></CENTER>
</BODY>
</HTML>
```

**הערה:**

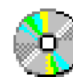
יש לשים לב שעליך לכלול את הביטוי =TARGET בערך המאפיין TargetFrame. זהו באג במימוש הנוכחי של הרכיב Ad Rotator ל- IIS 4.0.




**תרשים 9.5** שימוש במאפיין TargetFrame

# קובץ Rotator Schedule

לפני השימוש ברכיב Ad Rotator, יהיה עליך ליצור קובץ מיוחד הנקרא קובץ תזמון סבב (Rotator Schedule File). קובץ זה מכיל את כל המידע אודות כרזות הפרסומת. זהו קובץ טקסט רגיל, שתוכל ליצור ולערוך באמצעות כל עורך טקסט.



קובץ תזמון סבב מורכב משני חלקים. החלק הראשון מכיל מידע כללי לגבי כל הפרסומת שברצונך להציג. בחלק השני, מצוין מידע ספציפי לכל פרסומת. תדפיס 9.4 מראה דוגמה לקובץ זה:

**תדפיס 9.4** הקובץ ADROT.TXT

```
REDIRECT adredir.asp
WIDTH 468
HEIGHT 60
BORDER 0
*
aspsitead.gif
http://www.aspsite.com
The Active Server Pages Site
80
cityauctionad.gif
http://www.cityauction.com
CityAuction
20
```

שני קטעי המידע שבקובץ זה מופרדים האחד מהשני על ידי כוכבית (*). החלק הראשון מכיל ארבעה פרמטרים המשפיעים על כל הפרסומות בקובץ. הרשימה הבאה מסבירה את פעולתם של פרמטרים אלה:

- REDIRECT - פרמטר זה מציין קובץ הפניה (redirection file) לפרסומות. זהו הקובץ אליו ינותב המשתמש כאשר ילחץ על כרזת פרסומת.
- WIDTH - פרמטר זה מציין את רוחב תמונת הפרסומת בפיקסלים. ערך ברירת המחדל של פרמטר זה הוא 440 פיקסלים.
- HEIGHT - פרמטר זה מציין את גובה תמונת הפרסומת בפיקסלים. ערך ברירת המחדל של פרמטר זה הוא 60 פיקסלים.
- BORDER - פרמטר זה מציין את עובי הגבול שמסביב לתמונת הפרסומת. כברירת מחדל, עוביו של הגבול הוא פיקסל אחד.

בקובץ adrot.txt שבתדפיס 9.4, מצביע הפרמטר REDIRECT על הדף adredir.asp. הפרמטרים WIDTH ו- HEIGHT מציינים כי רוחבה של תמונת הפרסומת אמור להיות 468 פיקסלים, וגובהה אמור להיות 60 פיקסלים. לבסוף, ערך הפרמטר BORDER נקבע ל- 0, כך שלא יוצג כל גבול מסביב לתמונה.

הקטע השני מכיל מידע ייחודי לגבי כל פרסומת. בדוגמה זו, מכיל קובץ תזמון הסבב מידע לגבי שתי פרסומות. כרזת הפרסומת הראשונה היא של האתר ASPSite, ואילו הכרזה השנייה היא של אתר אינטרנט הנקרא CityAuction.

יש לספק ארבע שורות של מידע לגבי כל פרסומת. השורה הראשונה מציינת נתיב לתמונה שתוצג בפרסומת. תמונה זו יכולה להיות ממוקמת על השרת, או בכל מקום אחר ברשת האינטרנט.

השורה השנייה מכילה את כתובת ה-URL לאתר הבית של המפרסם. לאחר שמשתמש לחץ על הפרסומת, ניתן להפנותו לאתר זה (ראה בסעיף הבא). אם תציב מקף (-) בשורה זו, הפרסומת לא תתפקד כקישור.

השורה השלישית מציינת טקסט חלופי, שיוצג במידה ודפדפן המשתמש לא תומך בהצגת תמונות. שורה זו שקולה לתכונה (Attribute) ALT בתגית <IMAGE> של HTML. ניתן להציב בשורה כל מחרוזת טקסט רגילה שתרצה, בתנאי שאינה כוללת תגיות HTML.

### הערה:

לא ניתן להשתמש ברכיב Ad Rotator להצגת כרזות המכילות HTML דינמי. להצגת פרסומות מסוג זה (Rich Media), יש להשתמש ברכיב Content Rotator. רכיב זה נידון בפירוט בפרק 7.

לבסוף, השורה הרביעית קובעת את התכיפות בה תוצג פרסומת מסוימת. שורה זו מציינת את משקלה היחסי (Relative Weight) של הפרסומת ביחס לפרסומות האחרות. בדוגמה זו, תוצג הפרסומת הראשונה 80% מהפעמים, והפרסומת השנייה תוצג ב- 20% הנותרים.

### הערה:

על ידי קביעת משקל שונה לכל פרסומת, ניתן למכור פרסומות בתעריפים שונים. בדרך כלל, אתרי אינטרנט מוכרים פרסומות לפי מחיר לאלף צפיות (CPM, Cost Per Thousand). במילים אחרות, הקונים מחויבים בסכום מסוים הנובע ממספר הפעמים בהן הוצגו כרזות הפרסומת שלהם. הנוסחה המשמשת לקביעת העלות להצגת פרסומת היא:

Cost = (Number of Impressions / 1,000) x CPM

עלות = (מספר הפעמים שהכרזה הוצגה \ 1000) CPM X

לדוגמה, נניח וברצונך למכור באתר האינטרנט שלך שטח לכרזות פרסומת לפי CPM של 10.00 שקלים. אם קונה ירכוש ממך פרסומות בשווי של 100 שקלים, יהיה עליך להציג את הפרסומות שלו 10,000 פעמים. אם ממוצע הכניסות לאתר שלך הוא של 100,000 מבקרים בחודש, יהיה עליך להציג את כרזות הפרסומת של אותו הקונה ב 10% מהזמן. על ידי שימוש ברכיב Ad Rotator תוכל לקבוע כי על פרסומת להיות מוצגת בתדירות זו, על ידי קביעת משקלה היחסי של הפרסומת.

## קובץ Redirection

ניתן לציין קובץ הפניה המתייחס לכל הפרסומות בקובץ תזמון סבב (Rotator Schedule File) מסוים. משתמשים שילחצו על כרזת פרסומת יגיעו לקובץ זה. בדרך כלל זהו דף ASP, אך הוא יכול להיות גם דף HTML רגיל.

תפקידו העיקרי של קובץ זה הוא לעקוב אחר מספר הפעמים בהן נלחצה כרזת פרסומת מסוימת. לאחר שמידע זה נרשם, המשתמש מופנה בדרך כלל לאתר הבית של המפרסם. תדפיס 9.5 מציג דוגמה:

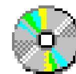
**תדפיס 9.5** הקובץ adredir.asp

```
<%
Response.AppendToLog Request.QueryString("url")
Response.Redirect Request.QueryString("url")
%>
```

קובץ הפניה זה מכיל תסריט ASP בן שתי שורות. השורה הראשונה רושמת לקובץ היומן (שם קובץ Log אשר מוגדר בשרת) מידע לגבי הפרסומת שנלחצה. לאחר מכן, נעשה שימוש בשיטה Redirect כדי לשלוח את המשתמש לאתר הבית של המפרסם.

בכל קריאה לקובץ ה הפניה, מועברות שתי מחרוזות שאילתה (Query Strings). מחרוזת השאילתה url מכילה את הנתיב לאתר הבית של המפרסם. זהו נתיב אתר הבית של המפרסם כפי שמופיע בקובץ תזמון הסבב.

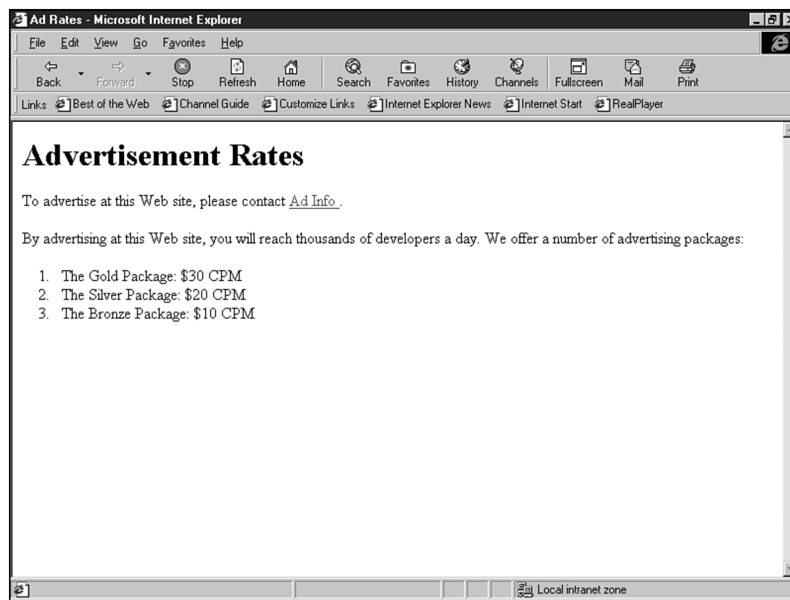
מחרוזת השאילתה השנייה נקראת image. מחרוזת שאילתה זו מכילה את נתיב תמונת הפרסומת שצוין בקובץ תזמון הסבב.



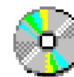
למעשה, ניתן לקבוע לקובץ ה הפניה כל תוכן שהוא. לדוגמה, תוכל להגדיר את קובץ ה הפניה לקובץ HTML רגיל, המציג את שערי הפרסום באתר שלך. תדפיס 9.6 מהווה דוגמה לקובץ כזה (ראה תרשים 9.6).

**תדפיס 9.6** adrates.asp

```
<HTML>
<HEAD><TITLE> Ad Rates </TITLE></HEAD>
<BODY>
<H1> Advertisement Rates </H1>
To advertise at this Web site, please contact
<A HREF="mailto:admaster@mysite.com"> Ad Info </A>.
<P>
By advertising at this Web site, you will reach
thousands of developers a day.
We offer a number of advertising packages:
<OL>
<LI>The Gold Package: $30 CPM
<LI>The Silver Package: $20 CPM
<LI>The Bronze Package: $10 CPM
</OL>
</BODY>
</HTML>
```




**תרשים 9.6** מידע לגבי פרסום



לעומת זאת, משמש בדרך כלל, קובץ ההפניה למדידת שיעור-ההכנסות (Click-through rate) לכרזות פרסומת. קיימות מספר שיטות לביצוע המדידה. ראשית, ניתן לשמור את הנתונים במשתני יישום (Application Variables). קובץ ההפניה בתדפיס 9.7 (adredirApp.asp) יוצר משתנה יישום חדש עבור כתובת URL של כל פרסומת.

**תדפיס 9.7** מעקב אחר פרסומות באמצעות משתני יישום

```
<%
  ' Get The Ad URL
  theURL = UCASE( Request.QueryString( "url" ) )
  ' Replace periods with addition signs
  theURL = "AD_" & REPLACE( theURL, ".", "+" )
  ' Increment the click-through count
  Application( theURL ) = Application( theURL ) + 1
  ' Redirect to the advertiser's page
  Response.Redirect Request.QueryString( "url" )
%>
```

תסריט זה מאחזר את מחרוזת השאילתה "url", המכילה את כתובת האתר של המפרסם. אחר כך, מומר ה-URL למשתנה VBScript תקין על ידי החלפת כל הנקודות שבו בסימני חיבור (+), והוספת המחרוזת "_AD" לפני שמו של המשתנה. לאחר מכן, משמש משתנה זה למניית שיעור הכניסות לפרסומת זו.

**הערה:**

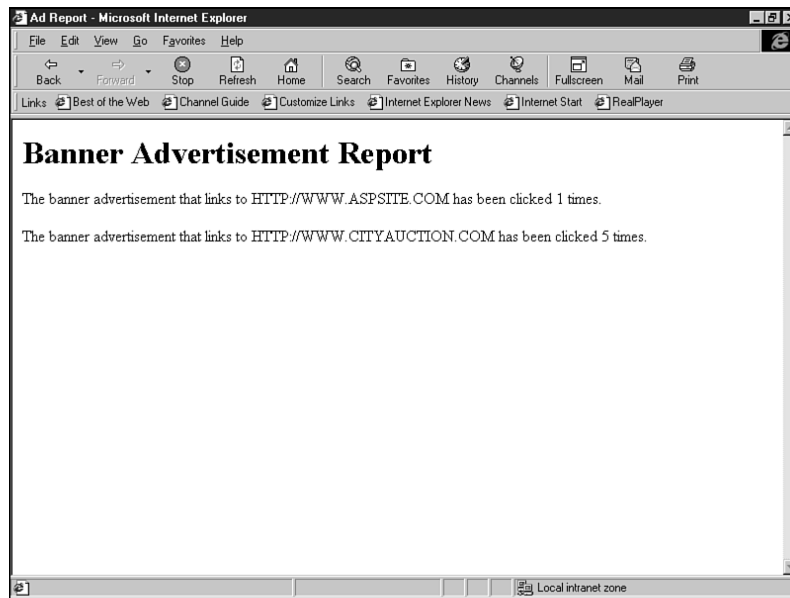
משתני VBScript תקינים צריכים להתחיל בתו של האלף-בית האנגלי, להכיל עד 255 תווים, ולא להכיל נקודות.

לאחר שמירת המידע במשתני היישום (Application Variables), ניתן ליצור דוח המפרט את מספר הכניסות לכל פרסומת, על ידי דף ASP שבתדפיס 9.8 (ראה תרשים *9.7*).

**תדפיס 9.8** adreportApp.asp

```
<HTML>
<HEAD><TITLE>Ad Report</TITLE></HEAD>
<BODY>
<H1>Banner Advertisement Report</H1>
<%
for each x in Application.Contents
 if LEFT( x, 3 ) = "AD_" then
 url = Replace(  x , "+", "." )
 url = RIGHT( url, LEN( url ) - 3 )
 clicks = Application( x )
 %>
 <P>
 The banner advertisement that links to <%=url%>
  has been clicked <%=clicks%> times.
 <%
 end if
next
%>
</BODY>
</HTML>
```

דף דיווח הפרסום (Advertisement Report Page) עובר על איברי האוסף Application בזה אחר זה, ומחפש משתני יישום ששמם מתחיל ב"_AD". ערך כל אחד ממשתנים אלה, המכיל את מספר הכניסות לפרסומת, מוצג בדפדפן.




**תרשים 9.7** דיווח הפרסום

גישה זו למעקב אחר לחיצות על פרסומות נוחה, אך יש לה חיסרון רציני. במקרה שבו שירות האינטרנט מושבת, הקובץ Global.asa עובר שינוי, או שאתר האינטרנט שלך קורס, יאבד כל המידע השמור במשתני היישום (Application Variables). אם עליך לדווח למפרסמים באתר שלך סטטיסטיקות אודות גישה לפרסומות, יהיה אבדן המידע הרה אסון.

כחלופה טובה יותר לשמירת נתונים לגבי מספר הלחיצות על כל פרסומת, תוכל להשתמש ברכיב Counters (ראה "מעקב אחר תצוגות דפים באמצעות הרכיב Counters", מוקדם יותר בפרק זה). כך תאבד כמות קטנה ביותר של נתונים במקרה של קריסת השרת, כיון שהנתונים נכתבים לקובץ טקסט באופן אוטומטי (ייתכן איבוד של נתונים מועטים, במקרה שקובץ הטקסט לא הספיק להתעדכן בזמן).

תדפיס 9.9 מהווה דוגמה לקובץ הפניה המשתמש ברכיב Counters (הקוד יוצא מנקודת ההנחה שמופע בשם myCount של הרכיב Counters נוצר בקובץ Global.asa).

**תדפיס 9.9** adredirCounters.asp

```
<%
 ' Get The Ad URL
 theURL = UCASE( Request.QueryString( "url" ) )
 ' Replace colons with addition signs
 theURL = "AD_" & REPLACE( theURL, ":", "+" )
 ' Increment the click-through count
 myCount.Increment( theURL )
 ' Redirect to the advertiser's page
 Response.Redirect Request.QueryString( "url" )
%>
```

חוץ מן השימוש ברכיב Counters במקום השימוש במשתני יישום (Application Variables), זהה קובץ הפניה זה לקובץ שבתדפיס 9.8. למרבה הצער לא כולל הרכיב Counters שיטה להחזרת שמות כל המונים, עובדה המקשה על הצגה אוטומטית של דוח הפרסום. לקבלת ערכי המונים, עליך להשתמש ברכיב File Access (ראה תדפיס 9.10).

**הערה:**

למידע נוסף אודות הרכיב File Access, ראה פרק 8.

**תדפיס 9.10** adreportCounters.asp

```
<%
Const counterFile = "c:\winnt\system32\inetsrv\counters.txt"
SET fs = Server.CreateObject( "Scripting.FileSystemObject" )
SET ts =  fs.OpenTextFile( counterFile )
%>
<HTML>
<HEAD><TITLE>Counters Ad Report</TITLE></HEAD>
<BODY>
<H1>Banner Advertisement Report</H1>
<%
'  Retrieve each counter
WHILE NOT ts.AtEndOfStream
 textLine = ts.ReadLine
 counterName = LEFT( textLine, INSTR( textLine, ":" ) - 1 )
 counterValue = RIGHT( textLine, LEN( textline ) - INSTRrev( textLine, ":" ) )
 if LEFT( counterName, 3 ) = "AD_" then
 url = Replace(  counterName , "+", ":" )
 url = RIGHT( url, LEN( url ) - 3 )
 clicks = counterValue
 %>
```

```
<P>
 The banner advertisement that links to <%=url%>
  has been clicked <%=clicks%> times.
 <%
 end if
WEND
%>
</BODY>
</HTML>
```

תסריט זה משתמש ברכיב File Access לפתיחת הקובץ Counters.txt. הרכיב Counters כותב את שמות המונים וערכיהם לקובץ זה באופן אוטומטי. לאחר מכן, נקראות שורות הקובץ בזו אחר זו, ומכל שורה מחולצים שם המונה וערכו. אם שמו של המונה מתחיל ב"_AD", אז שמו וערכו מוצגים בדפדפן.

**הערה:**

לפני שתוכל להשתמש ברכיב File Access לקריאת הקובץ Counters.txt, ייתכן ותצטרך להגדיר עבור החשבון IUSER_MachineName הרשאות קריאה (Read Permissions) לקובץ זה. אם אינך יודע כיצד לעשות זאת, פנה לפרק 8.

# סיכום

בפרק זה למדת כיצד לעקוב אחר מבקרים באתר ולהציג פרסומות. למדת לנהל מעקב אחר המבקרים באתר שלך על ידי שימוש ברכיב Web Counter וברכיב Counters. כמו כן, למדת לשלב פרסומות בדפי האינטרנט שלך, באמצעות הרכיב Ad Rotator. לבסוף, נידונו שתי דרכים למעקב אחר שיעור הכניסות לפרסומות. למדת כיצד לנהל מעקב אחר כניסות לפרסומות באמצעות משתני יישום, ובאמצעות הרכיב Counters.

## פרק 10

# עבודה עם דואר אלקטרוני

**בפרק זה:**

- **שירות SMTP של Microsoft**
- **משלוח דואר אלקטרוני באמצעות CDO עבור NTS**
- **אחזור דואר אלקטרוני עם CDO עבור NTS**
- **יצירת Simple List Server**

במצבים רבים תרצה לשלוח הודעת דואר אלקטרוני (E-Mail) מאתר האינטרנט שלך באופן אוטומטי. לדוגמה, לאחר רישום (Registration) משתמש לאתר, ייתכן ותרצה לשלוח לו באופן אוטומטי הודעת דואר אלקטרוני עם פרטי הרישום שביצע. או, אם האתר שלך משמש למכירת תוכנה, ייתכן ותרצה לשלוח את התוכנה כקובץ המצורף להודעת דואר אלקטרוני; אם למשל בבעלותך אתר חדשות והינך מעוניין לשלוח לנרשמים לאתר את החדשות האחרונות, תצטרך לדעת איך לשלוח E-Mail דרך ASP.

בפרק זה, תלמד כיצד לשלוח ולאחזר הודעות דואר אלקטרוני מתוך דפי ASP שלך. נתמקד במיוחד באופן השימוש ב-Collaboration Data Objects (אובייקטים לשיתוף נתונים) עבור שרתי Windows NT (הידועים בקיצור כ-CDO for NTS או CDO עבור NTS), למטרת תקשור עם שירות SMTP (Simple Mail Transfer Protocol) של Microsoft.

בגרסאות IIS (Internet Information Server) הקודמות לגירסה 4.0, היה תהליך משלוח הודעות דואר אלקטרוני מדף ASP מסובך. לא הייתה ברירה מלבד קנייה של רכיב צד שלישי לטיפול בדואר אלקטרוני, או כתיבה עצמאית של רכיב כזה. למרבה המזל, כללה Microsoft ב-Option Pack של Windows NT 4.0 גם שרת SMTP, וכן אובייקטים רבים לתקשור עם שרתי דואר ב- IIS 4.0.

בחלקו הראשון של פרק זה, תלמד כיצד לקבוע את התצורה של שירות SMTP, וכיצד להשתמש בשירות זה. שירות SMTP הוא שאחראי על ביצוע העבודה הכרוכה בתחלופת הודעות הדואר האלקטרוני עם העולם החיצון.

חלקו השני של פרק זה מכסה את השיטות לשימוש ב-CDO for NTS למשלוח דואר אלקטרוני. תלמד לבנות שגרה (Subroutine) פשוטה, שתאפשר לך לשלוח דואר אלקטרוני באמצעות שימוש במשפט אחד בלבד בתסריט ASP. בנוסף, תלמד כיצד להשתמש במספר מאפיינים מתקדמים של CDO for NTS, שיאפשרו לך לשלוח הודעות דואר אלקטרוני ולצרף להן קבצים, לקבוע להן עדיפויות שונות, וכן לכלול עיצובי HTML ותמונות.

בחלק השלישי, תלמד כיצד לגשת לספריה Drop (Drop directory) לאחזור הודעות דואר אלקטרוני שנשלחו לאתר שלך. בחלק זה, תלמד את השיטות המשמשות לאחזור ומיון הודעות הדואר האלקטרוני שנכנסו.

לבסוף, בחלקו האחרון של הפרק, תלמד כיצד להשתמש ב-CDO for NTS ובשירות SMTP ליצירת שרת רשימה פשוט. שרת זה יוכל לשמש ליצירת קבוצת דיון מבוססת דואר אלקטרוני ולתחזוקתה.

# שירות SMTP של Microsoft

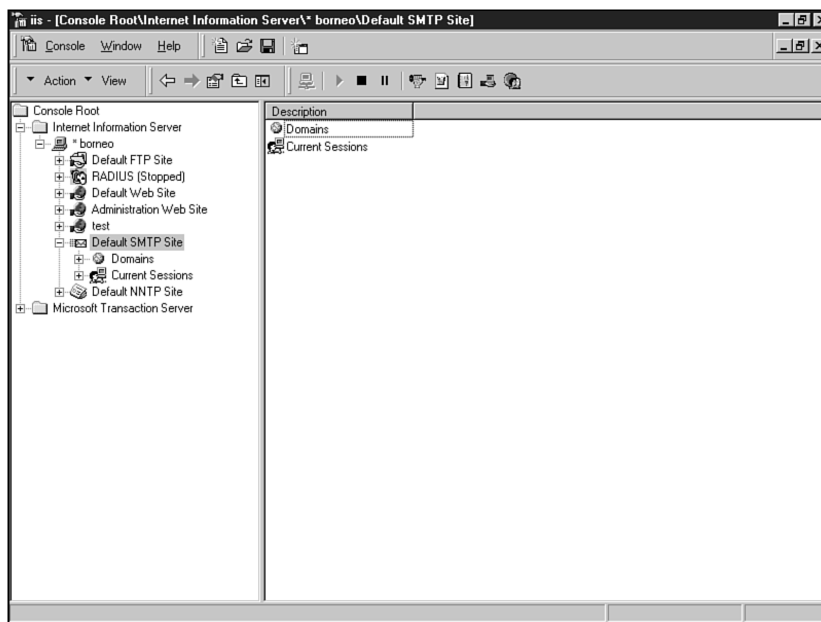
שירות SMTP של Microsoft אחראי על הפעולה הבסיסית של משלוח ואחזור של הודעות דואר אלקטרוני בתיאום עם פרוטוקול SMTP. במובנים מסוימים, זהו שירות מוגבל מאד. השירות אינו תומך בפרוטוקול Post Office (POP). פרוש הדבר שלא ניתן להשתמש בשירות ליצירת חשבונות תיבת-דואר למספר משתמשים, או לאחזור דואר מהשירות על ידי שימוש בלקוחות דואר אלקטרוני דוגמת Outlook Express או Eudora.

ליצירת מערכת דואר אלקטרוני מלאה, התומכת בפעולות שמעבר ל-SMTP בסיסי, יהיה עליך להשקיע בתוכנה נוספת, דוגמת Microsoft Exchange Server, NTMail של Interner Shopper מהכתובת http://www.ntmail.co.uk) או Post.Office של Software.com (בכתובת http://www.software.com).

אבל, בצד החיובי, השירות שימושי מאד למשלוח הודעות דואר אלקטרוני אוטומטיות מאתר האינטרנט שלך. השירות תומך במשלוח של אלפי הודעות דואר אלקטרוני ליום. לדוגמה, תוכל להשתמש בו לניהול מסע פרסום מבוסס דואר אלקטרוני. כמו כן תוכל להשתמש בשירות ליצירת מספר אתרי SMTP התומכים במספר Domains, על שרת יחיד.

# קביעת התצורה של שירות SMTP

כדי לבדוק אם שירות SMTP מותקן ופעיל בשרת שלך, פתח את Internet Service Manager , ובדוק האם הסמל Default SMTP Site מופיע בתיקיה Internet Information Server (ראה תרשים 10.1). אם הסמל אינו מופיע, עליך להתקין את השירות על ידי הפעלת תוכנית ההתקנה של ה-Option Pack עבור Windows NT. אם השירות מותקן אך אינו פעיל (Stopped), תוכל להפעילו על ידי בחירה באפשרות Start מתפריט Action.




**תרשים 10.1** גישה לשירות SMTP דרך MMC.

בהתקנת שירות SMTP, נוצר Domain ברירת מחדל מקומי (Local Default Domain). שמו של Domain ברירת המחדל המקומי קובע כיצד יעביר השירות הודעות. אם השירות יקבל הודעת דואר אלקטרוני הממוענת ל-Domain הזהה ל-Domain ברירת המחדל המקומי, תועבר ההודעה באופן מקומי. אחרת, ינסה השירות לשלוח את ההודעה למקום אחר. שמו של Domain ברירת המחדל המקומי זהה לשמו של אתר האינטרנט שלך.

תוכל ליצור כינויים (Alias domains) נוספים, שייצגו domains שונים באתר SMTP יחיד. כינויי domain שימושיים כאשר לאתר האינטרנט שלך מספר שמות domain שונים. ליצירת Alias domain, פתח את Internet Service Manager, לחץ לחיצה ימנית על אתר ברירת המחדל (Default SMTP Site) שלך, פתח את תפריט New ובחר Domain. על כל Alias domains הנוספים להיות תחת אותו default local domain בספריית השיתוף (Share Directory) (ראה בסעיף הבא).

תוכל אפילו ליצור מספר אתרי SMTP באותו השרת. כדי לעשות כן, עליך לוודא כי ברשותך הגירסה האחרונה של שירות SMTP. נכון להיום, כולל ה-Option Pack של Windows NT את גירסה 1.0 של שירות SMTP. גירסה חדשה של שירות SMTP נמצאת ב-Service Pack 4. כדי ללמוד כיצד להתקין את הגירסה החדשה וליצור אתרי SMTP נוספים באותו השרת, חפש את המאמר שמספרו Q183476 באתר האינטרנט http://www.microsoft.com/search.

לאחר התקנת שירות SMTP, תוכל לעקוב אחר ביצועיו על ידי עיון בקבצי היומן (Log). כברירת מחדל, שומר השירות יומן של פעולותיו בתיקיה \WINNT\System32\LogFiles. ניתן לשנות תיקיה זו על ידי עריכת הערך Log File Directory שבמאפייני השירות. ניתן לפתוח ולעיין בקבצי היומן (Log) של שירות SMTP באמצעות כל עורך טקסט תקני, דוגמת Notepad.

# כיצד פועל שירות SMTP

מנקודת מבטו של המשתמש, שירות SMTP הוא רכיב פשוט מאד. הרכיב משתמש בשתי ספריות עיקריות (Main Directories) בשם Pickup ו-Drop לעיבוד דואר אלקטרוני. שתי הספריות ממוקמות תחת הספריה InetPub\MailRoot.

הספריה Pickup משמשת למשלוח דואר אלקטרוני. השירות מנטר באופן קבוע את הספריה ומחפש הודעות דואר אלקטרוני חדשות, וכשהוא מוצא הודעה חדשה הוא מנסה לשלוח אותה. אם השירות לא מסוגל להעביר את ההודעה באופן מיידי, היא נשמרת בספריית התור (Queue Directory) בעוד השירות ממשיך לנסות ולהעבירה. אם לא ניתן להעביר את הודעת הדואר האלקטרוני, וגם לא ניתן להחזירה לשולח, מועברת ההודעה לספריית הדואר הפגום (BadMail Directory).

הספריה Drop משמשת לקבלת דואר אלקטרוני. כשהודעות דואר אלקטרוני מתקבלות על ידי שירות SMTP, הן ממוקמות בספריה Drop. השירות לא עושה עם ההודעות שום דבר אחר. הן פשוט נשארות שם. נציין שוב, כי שירות SMTP של Microsoft אינו תומך ביותר מתיבת דואר אחת.

הודעות הדואר האלקטרוני שבשתי ספריות אלו הן קבצי טקסט פשוטים. לדוגמה, באפשרותך לשלוח הודעת דואר אלקטרוני על ידי פתיחת Notepad, הקלדת הטקסט הבא, ושמירת הקובץ בספריה Pickup (Inetpub\Mail\pickup):

```
To: webmaster@aspsite.com
From: someone@somewhere.com
Subject: Testing SMTP Service

Here is the message!
```

הקובץ ייעלם מהספריה Pickup מייד לאחר השמירה, כיון ששירות SMTP מנסה לשלחו.

באופן תיאורטי, ניתן להשתמש ברכיב File Access מתוך דפי ASP כדי לכתוב הודעות לספריה Pickup, ועל ידי כך לשלוח דואר אלקטרוני (למידע נוסף על הרכיב File Access, ראה פרק 8). באופן מעשי, אין זו דרך הפעולה הטובה ביותר, משתי סיבות. ראשית, מפעם לפעם גורמת שיטה זו למשלוח הודעות חלקיות בלבד (נראה כי שירות SMTP מתעלה על עצמו ושולח את הדואר האלקטרוני לפני סיום כתיבת הקובץ לתיקיה). שנית, לא קל להשתמש בשיטה זו בעבודה עם אפשרויות דואר אלקטרוני מתקדמות, כגון עדיפויות (Priorities) וצירוף קבצים (File Attachments).

השיטה האמינה והקלה ביותר למשלוח דואר אלקטרוני באמצעות שירות SMTP היא שימוש ב-CDO (Collaboration Data Objects). שאר הפרק יוקדש ללימוד דרך פעולה זו.

# משלוח דואר אלקטרוני באמצעות CDO for NTS

קל מאד להשתמש ב-CDO למשלוח הודעת דואר אלקטרוני מתוך דף ASP. ניתן לבצע זאת על ידי שגרה בת שמונה משפטים בלבד. לפניך דוגמה:

```
SUB SendMail (fromWho, toWho, Subject, Body)
    DIM myMail
    SET myMail = Server.CreateObject("CDONTS.Newmail")
    myMail.From = fromWho
    myMail.To = toWho
    myMail.Subject = Subject
    myMail.Body = Body
    myMail.Send
    SET myMail = Nothing
END SUB
```

שגרה זו מקבלת ארבעה פרמטרים :

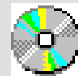
- כתובת הדואר האלקטרוני של שולח ההודעה.
- כתובת הדואר האלקטרוני של נמען ההודעה.
- נושא ההודעה.
- גוף ההודעה.

תחילה נוצר מופע של האובייקט NewMail, לאחר מכן נקבעים המאפיינים המתאימים, ולבסוף נעשה שימוש בשיטה Send כדי לשלוח את הודעת הדואר האלקטרוני.

עליך לקבוע את כתובת השולח בהודעות היוצאות לכתובת דואר אלקטרוני ב-Domain שלך. אם לא תעשה כן, עלולים שרתי דואר אלקטרוני אחרים לדחות את ההודעות שלך באופן אוטומטי.

תדפיס 10.1 הוא דף ASP פשוט הכולל טופס (Form) למשלוח דואר אלקטרוני. הטופס מופיע בתרשים 10.2.

**הערה:**

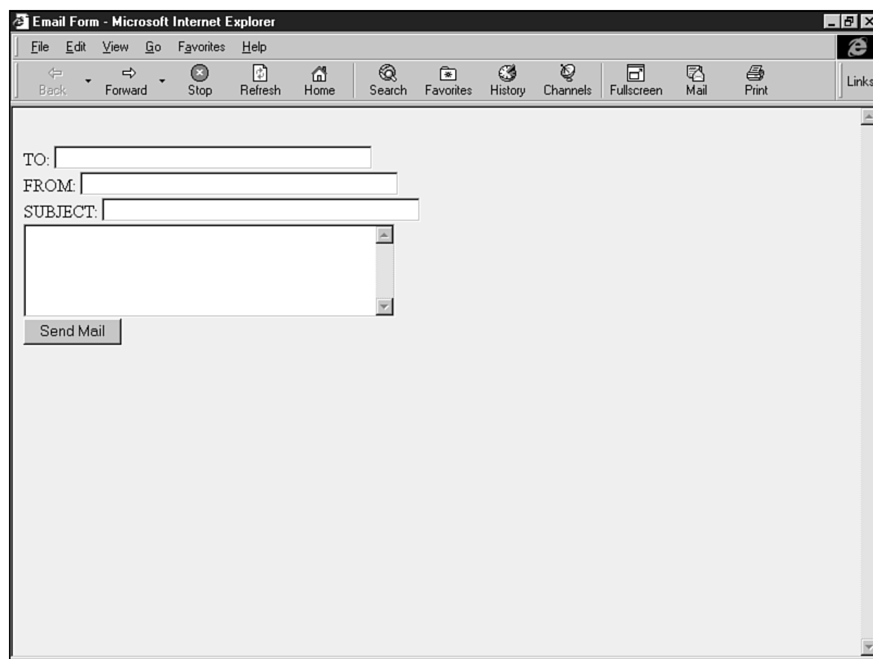


ניתן למצוא את טופס משלוח הדואר (Send Mail) בקובץ sendmail.asp שבתקליטור המצורף לספר זה.

**תדפיס 10.1** טופס משלוח הדואר

```
<%
SUB sendMail( fromWho, toWho, Subject, Body )
    DIM myMail
    SET myMail = Server.CreateObject("CDONTS.Newmail")
    myMail.From = fromWho
    myMail.To = toWho
    myMail.Subject = Subject
    myMail.Body = Body
    myMail.Send
    SET myMail = Nothing
END SUB

fromWho = TRIM( Request.Form( "fromWho" ) )
toWho = TRIM( Request.Form( "toWho" ) )
Subject = TRIM( Request.Form( "Subject" ) )
Body = TRIM( Request.Form( "Body" ) )
```

```
IF toWho <> "" THEN
    sendMail fromWho, toWho, Subject, Body
END IF
%>
<HTML>
<HEAD><TITLE> E-Mail Form </TITLE></HEAD>
<BODY BGCOLOR="#eeeeee">

<FORM METHOD="POST"
ACTION="<%=Request.ServerVariables("SCRIPT_NAME")%>">
<BR>TO: <INPUT NAME="toWho" TYPE="text" SIZE=40>
<BR>FROM: <INPUT NAME="fromWho" TYPE="text" SIZE=40>
<BR>SUBJECT: <INPUT NAME="subject" TYPE="text" SIZE=40>
<BR><TEXTAREA NAME="Body" COLS=40 ROWS=5></TEXTAREA>
<BR><INPUT TYPE="SUBMIT" VALUE="Send Mail">
</FORM>

</BODY>
</HTML>
```




**תרשים 10.2** טופס משלוח הדואר

**טיפ:**

במקום לקבוע כל אחד ממאפייני האובייקט NewMail בנפרד, תוכל לקבוע את כולם במשפט יחיד בתוך השיטה Send. לדוגמה, המשפט הבא שולח הודעת דואר אלקטרוני:

```
myMail.Send("sender@somewhere.com", "recipient@somewhere.com",
    ↳ "Here is the subject", "Hello!")
```

במקרה של משלוח הודעה ארוכה, בה גוף ההודעה כולל מעברי שורה (Line Breaks), תוכל להשתמש בקבוע vbCRLF. זהו קבוע של VBScript, המוסיף תו Enter (Carriage Return) והזנת שורה (Line Feed). להלן דוגמה של הודעת דואר אלקטרוני בעלת מספר שורות:

```
DIM theMessage
theMessage = "This is the first line." & vbCRLF &
        ↳"This is the second line." & vbCRLF &
        ↳"This is the third line."
SET myMail = Server.CreateObject("CDONTS.Newmail")
myMail.MailFormat = 0
myMail.BodyFormat = 0
myMail.From = "server@yourdomain.com"
myMail.To = "user@somewhere.com"
myMail.Subject = "Welcome to our Web site!"
myMail.Body = theMessage
myMail.Send
SET myMail = NOTHING
```

בעת משלוח דואר אלקטרוני, תוכל לשלוח את ההודעה למספר חשבונות דואר אלקטרוני שונים. כדי להפריד בין הנמענים השונים הקלד תו נקודה-פסיק (;) בין כתובת אחת לשנייה. המשפט הבא מציין כי הודעה מסוימת תשלח לכל חשבונות הדואר האלקטרוני שברשימה:

```
myMail.To = "user1@somewhere.com; user2@somewhere.com;
    ↳ user3@somewhere.com"
```

שים לב, כשאתה משתמש באובייקט NewMail למשלוח דואר אלקטרוני, עליך ליצור מופע חדש של האובייקט לכל הודעת דואר אלקטרוני שתשלח. לדוגמה, קטע הקוד הבא מוביל לשגיאת הרצה:

```
DIM objMail, k
SET myMail = Server.CreateObject("CDONTS.Newmail")
FOR k = 1 TO 50
    myMail.From = "sender@somewhere.com"
    myMail.To = "recipient@somewhere.com"
    myMail.Subject = "message #" & k
    myMail.Body = "Test Message"
    myMail.Send
NEXT
```

אם תרצה לשלוח מספר הודעות דואר אלקטרוני מתוך דף ASP יחיד, יהיה עליך ליצור ולפרק (Destroy) מספר מופעים של האובייקט, כדי להימנע מקבלת השגיאה. התסריט הבא שולח מספר הודעות דואר אלקטרוני מבלי להיכשל, על ידי יצירת אובייקט NewMail חדש עבור כל הודעה, ולאחר מכן פירוק כל אובייקט על ידי קביעת ערכו ל- Nothing :

```
DIM objMail, k
SET myMail = Server.CreateObject("CDONTS.Newmail")
FOR k = 1 TO 50
    myMail.From = "test@somewhere.com"
    myMail.To = "test@somewhere.com"
    myMail.Subject = "message #" & k
    myMail.Body = "Test Message"
    myMail.Send
    SET myMail = NOTHING
NEXT
```

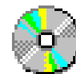
מגבלה זו אינה באמת מגבלה. דף ASP יחיד מסוגל לשלוח אלפי הודעות דואר אלקטרוני שונות, על ידי יצירת האובייקט NewMail ופירוקו שוב ושוב.

# צירוף קבצים להודעות דואר אלקטרוני

ניתן לצרף קבצים להודעת דואר אלקטרוני על ידי שימוש בשיטה ()AttachFile של האובייקט NewMail. השיטה ()AttachFile מקבלת שלושה פרמטרים : שמו ונתיבו המלאים של הקובץ הנשלח, שמו של הקובץ כפי שיופיע בהודעת הדואר האלקטרוני, והשיטה בה יקודד הקובץ לתוך ההודעה :

objNewMail.**AttachFile**(Source [, FileName] [, EncodingMethod] )

כפי שניתן לראות, שני הפרמטרים האחרונים אופציונליים בלבד. הפרמטר Source מציין את שמו ונתיבו המלאים של הקובץ המצורף להודעה. הפרמטר Filename הוא שמו של הקובץ שיופיע בהודעה – במקרה שפרמטר זה חסר, יופיעו בהודעה שמו ונתיבו המלאים של הקובץ. ערכו של הפרמטר EncodingMethod קובע את שיטת קידוד הקובץ : עבור הערך 0 יעשה שימוש בקידוד UUEncode, ועבור הערך 1 יעשה שימוש בקידוד Base 64.



לדוגמה, דף ASP שבתדפיס 10.2 ישלח את עצמו כקובץ המצורף להודעה באופן אוטומטי, לאחר שהמשתמש יכניס את כתובתו וילחץ על Send Source.

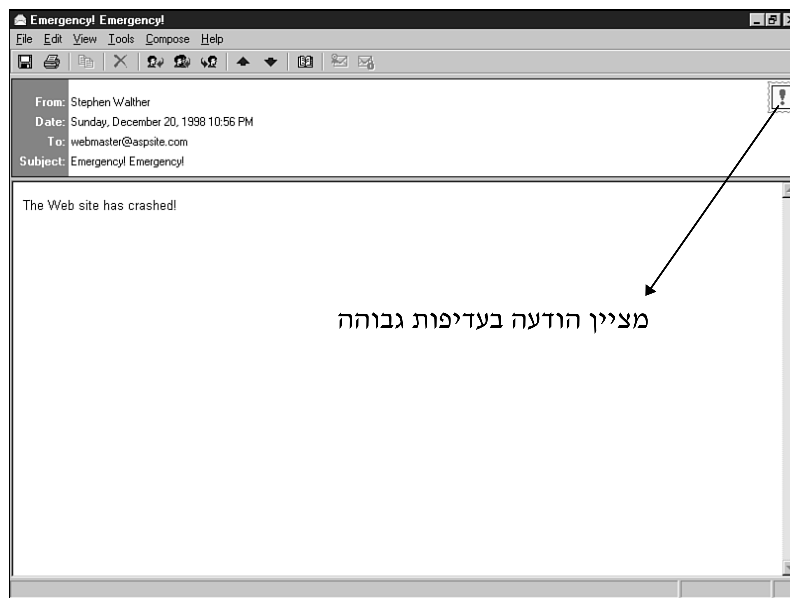
**תדפיס 10.2** משלוח קוד המקור של הקובץ (SendSource.asp)

```
<%
DIM myMail, thisPage, email
thisPage = Request.ServerVariables( "PATH_TRANSLATED" )
email = TRIM( Request.Form( "email" ) )
IF email <> "" THEN
    SET myMail = Server.CreateObject("CDONTS.Newmail")
    myMail.From = "source@yourdomain.com"
```

```
    myMail.To = email
    myMail.Subject = "Source for " & thisPage
    myMail.Body = "This email has the source for file " & thisPage
    myMail.AttachFile ThisPage, "Source"
    myMail.Send
    SET myMail = NOTHING
END IF
%>
<HTML>
<HEAD><TITLE>Some Page</TITLE></HEAD>
<BODY>
<FORM METHOD="POST" ACTION="sendsource.asp">
Email Address: <INPUT NAME="email" TYPE="TEXT" SIZE=40>
<INPUT TYPE="SUBMIT" VALUE="Send Source">
</FORM>
</BODY>
</HTML>
```

# קביעת עדיפויות בהודעות דואר אלקטרוני

מרבית לקוחות הדואר האלקטרוני תומכים בעדיפויות (Priorities) של הודעות. לדוגמה, Microsoft Outlook מאפשרת משלוח הודעות בעדיפויות Low, Normal או High. כשמתקבלת הודעה בעלת עדיפות גבוהה (High Priority), היא מוצגת כמתואר בתרשים 10.3. indicate




**תרשים 10.3** קבלת הודעה חשובה

ניתן להשתמש במאפיין Importance של האובייקט NewMail לקביעת עדיפות ההודעה. ניתן לקבוע את ערך המאפיין לאחד מהבאים:

0 - מציין חשיבות נמוכה

1 - מציין חשיבות רגילה (זהו גם ערך ברירת המחדל)

2 - מציין חשיבות גבוהה.

התסריט הבא שולח הודעת דואר אלקטרוני בעלת חשיבות גבוהה:

```
<%
DIM myMail
SET myMail = Server.CreateObject("CDONTS.Newmail")
myMail.From = "server@yourdomain.com"
myMail.To = "administrator@yourdomain.com"
myMail.Subject = "Web Site Crashed!!!"
myMail.Body = "The Web Site has crashed!"
myMail.Importance = 2
myMail.Send
SET myMail = NOTHING
%>
```

# משלוח CC ו-BCC

כששולחים הודעת דואר אלקטרוני, ניתן לציין כי עותק מההודעה יישלח לכתובת דואר אלקטרוני נוספת. קיימות שתי דרכים למשלוח עותק של הודעה. כששולחים הודעה עם עותק (CC, Carbon Copy), יכול הנמען הראשי לראות את כתובות הדואר האלקטרוני אליהן נשלחו העותקים. לפניך דוגמה לתסריט השולח עותק רגיל:

```
<%
DIM myMail
SET myMail = Server.CreateObject("CDONTS.Newmail")
myMail.From = "fre@aol.com"
myMail.To = "alice@yahoo.com"
myMail.Subject = "Confidential Information"
myMail.Body = "Alice, I haven't sent any email to Bob"
myMail.Cc = "bob@ix.netcom.com"
myMail.Send
SET myMail = NOTHING
%>
```

בדוגמה זו, משמש המאפיין Cc של האובייקט NewMail למשלוח עותק של הודעת הדואר האלקטרוני לחשבון הדואר האלקטרוני alice@yahoo.com.

ניתן לשלוח גם עותק נסתר (Blind Carbon Copy ,BCC). כששולחים הודעה עם עותק נסתר, לא ידע הנמען הראשי כי נשלחו עותקים נוספים של ההודעה. המאפיין Bcc של האובייקט NewMail משמש למשלוח עותקים נסתרים.

```
<%
DIM myMail
SET myMail = Server.CreateObject("CDONTS.Newmail")
myMail.From = "fred@aol.com"
myMail.To = "alice@yahoo.com"
myMail.Subject = "Confidential Information"
myMail.Body = "Alice, I haven't sent any email to Bob"
myMail.Bcc = "bob@ix.netcom.com"
myMail.Send
SET myMail = NOTHING
%>
```

משלוח עותקים נסתרים יכול להיות שימושי מאד למעקב אחר פעילות הדואר האלקטרוני המתרחשת באתר האינטרנט שלך. לדוגמה, נניח והאתר שלך שולח אוטומטית לכל משתמש חדש הודעות דואר אלקטרוני עם מידע הרישום. תוכל לשלוח עותק של ההודעות לחשבון דואר אלקטרוני נפרד, כדי לבדוק אם הן אכן נשלחות, או כדי לשמור באופן אוטומטי עותק של כל נתוני הרישום.

תוכל לציין יותר מחשבון דואר אלקטרוני אחד כנמען, גם בעותק רגיל וגם בעותק נסתר. כדי לעשות זאת, כל שעליך לעשות הוא להפריד בין כתובות הדואר האלקטרוני השונות באמצעות התו נקודה-פסיק (;), כך:

```
myMail.Bcc = "user1@somewhere.com; user2@somewhere.com;
   ↳ user3@somewhere.com"
```

# משלוח דואר אלקטרוני עם עיצובי HTML ותמונות

ניתן להשתמש ב-CDO למשלוח הודעות הכוללות עיצובי HTML ותמונות. לדוגמה, תוכל לכלול טקסט בגופנים ובצבעים שונים, לקבוע תמונת רקע מותאמת אישית, ולהוסיף קישורי HTML.

אולם, כדאי שתדע כי לא כל לקוחות הדואר האלקטרוני תומכים באופן מלא בעיצובי HTML ובתמונות. למרות שתוכנת Outlook של Microsoft תומכת באפשרויות אלו, משלוח דואר אלקטרוני בעיצוב HTML, ללקוחות דואר אלקטרוני אחרים עלולה לגרום להודעה להתקבל כקשקוש לא קריא במיוחד, שכן יוצג קוד המקור של HTML בלבד.

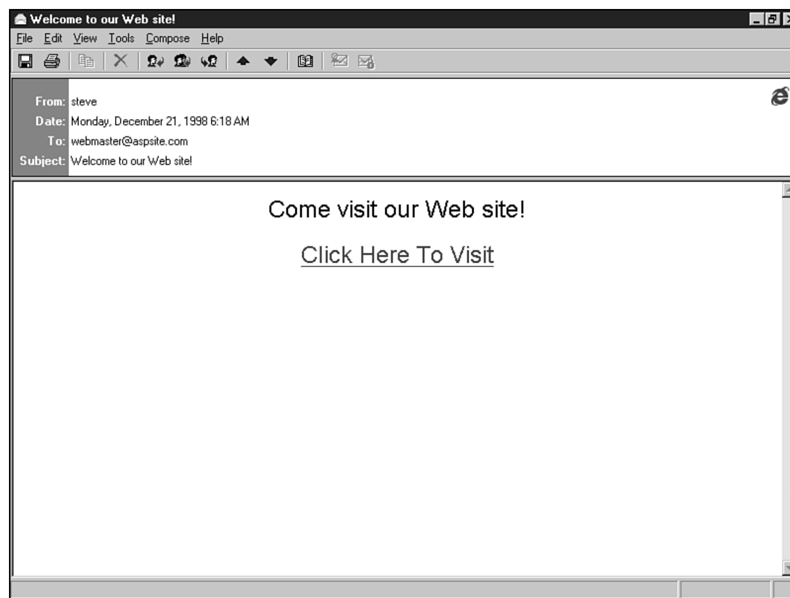
כברירת מחדל, שולח CDO הודעות דואר אלקטרוני כטקסט פשוט (Plain Text). כדי לשלוח דואר אלקטרוני המכיל HTML, עליך לקבוע ערכים לשניים ממאפייני האובייקט NewMail: המאפיין MailFormat והמאפיין BodyFormat:

```
myMail.MailFormat = 0
myMail.BodyFormat = 0
```

כשערך המאפיין MailFormat נקבע ל- 0, יישלח הדואר האלקטרוני בפורמט MIME.
כשערך המאפיין BodyFormat נקבע ל- 0, יכולה ההודעה להכיל HTML. לאחר קביעת ערכם של שני מאפיינים אלה, תוכל לשלוח הודעת דואר אלקטרוני בעיצוב HTML כך:

```
<%
DIM myMail, HTML
HTML = "<CENTER><FONT SIZE=+2 FACE=""Ariel""> " &
   ↳"Come visit our Web Site!"  &
   ↳"<P><A HREF=""http://www.yourdomain.com"">
   ↳ Click Here to Visit</A>" &
   ↳"</FONT></CENTER>
SET myMail = Server.CreateObject("CDONTS.Newmail")
myMail.MailFormat = 0
myMail.BodyFormat = 0
myMail.From = "server@yourdomain.com"
myMail.To = "user@somewhere.com"
myMail.Subject = "Welcome to our Web Site!"
myMail.Body = HTML
myMail.Send
SET myMail = NOTHING
%>
```

תסריט זה שולח הודעת דואר אלקטרוני בה גוף ההודעה מופיע בגופן גדול מסוג Ariel. ההודעה שבדוגמה זו כוללת גם קישור HTML, המוביל לאתר האינטרנט שכתובתו http://www.yourdomain.com (ראה תרשים 10.4).




**תרשים 10.4** דואר אלקטרוני עם קישור HTML.

כשאתה כולל קישורים בהודעת דואר אלקטרוני, עליך לכלול את הכתובת המדויקת והמלאה לקישור (אחרת, לא יהיו ללקוח הדואר האלקטרוני מספיק נתונים למציאת האתר). במילים אחרות, עליך להשתמש בכתובות במבנה http://www.yourdomain.com/thepage.asp, ולא בכתובת הקצרה יותר יחסית thepage.asp. תוכל להתגבר על מגבלה זו על ידי שימוש במאפיין ContentLocation של האובייקט NewMail. להלן דוגמה לכך:

```
<%
SET myMail = Server.CreateObject("CDONTS.Newmail")
myMail.MailFormat = 0
myMail.BodyFormat = 0
myMail.ContentLocation = "http://www.yourdomain.com/"
myMail.From = "server@yourdomain.com"
myMail.To = "user@somewhere.com"
myMail.Subject = "Welcome to our Web Site!"
myMail.Body = "<A HREF=""thepage.asp"">Visit Our Web Site!</A>"
myMail.Send
SET myMail = NOTHING
%>
```

קיימות שתי שיטות לכלול תמונות בהודעת הדואר האלקטרוני. השיטה הקלה ביותר היא לכלול בגוף ההודעה את התגית <IMG> של HTML. התסריט הבא משתמש בשיטה זו להצגת תמונת GIF של דינוזאור (ראה תרשים 10.5):

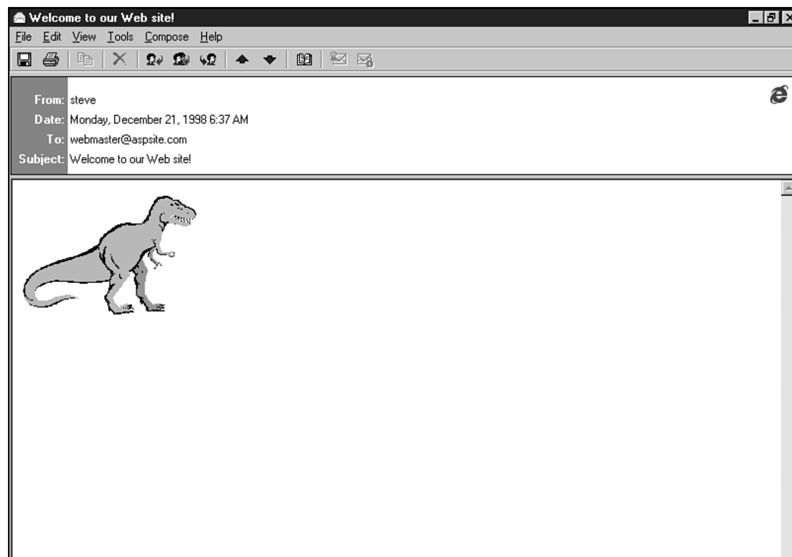
```
<%
SET myMail = Server.CreateObject("CDONTS.Newmail")
myMail.MailFormat = 0
myMail.BodyFormat = 0
myMail.ContentLocation = "http://www.yourdomain.com/images/"
myMail.From = "server@yourdomain.com"
myMail.To = "user@somewhere.com"
myMail.Subject = "Welcome to our Web Site!"
myMail.Body = "<IMG SRC=""dinosaur.gif"">"
myMail.Send
SET myMail = NOTHING
%>
```

בעת צפייה בהודעה זו, נמשכת התמונה משרת האינטרנט. אם השרת יפסיק את פעולתו, או שקובץ התמונה ימחק מהשרת, לא ניתן יהיה לצפות בתמונה. כדי לעקוף בעיה זו, תוכל לשלוח את התמונה עצמה עם הודעת הדואר האלקטרוני. כדי לעשות כן, השתמש בשיטה ()AttachURL של האובייקט NewMail:

```
<%
SET myMail = Server.CreateObject("CDONTS.Newmail")
myMail.MailFormat = 0
myMail.BodyFormat = 0
myMail.AttachURL "d:\inetpub\wwwroot\images\dinosaur.gif", "dinosaur.gif"
myMail.ContentLocation = "http://www.yourdomain.com/images/"
myMail.From = "server@yourdomain.com"
myMail.To = "user@somewhere.com"
```

```
myMail.Subject = "Welcome to our Web Site!"
myMail.Body = "<IMG SRC=""dinosaur.gif"">"
myMail.Send
SET myMail = NOTHING
%>
```

בקטע קוד זה, מקבלת השיטה ()AttachURL את שמו ונתיבו המלאים של קובץ התמונה. היות והתמונה מוכלת בהודעת הדואר האלקטרוני, היא מוצגת באופן מיידי. תוכל לצרף תמונות נוספות על ידי קריאות חוזרות לשיטה ()AttachURL.




**תרשים 10.5** דואר אלקטרוני עם תמונה.

# אחזור דואר אלקטרוני באמצעות CDO for NTS

כשהודעות דואר אלקטרוני מגיעות לאתר האינטרנט שלך, מפנה אותן שירות SMTP לספריה Drop. אם תרצה לצפות בהודעות אלו, דבר לא יעצור אותך מלהשתמש בפנקס הרשימות (Notepad) כדי לפתוח ולקרוא אותן.

הודעות הדואר האלקטרוני הן בסך הכל קבצי טקסט רגילים.

תוכל להשתמש ב-CDO for NTS לאחזור ההודעות מהספריה Drop, מתוך תסריט דפי ASP שלך. כדי לעשות זאת, תצטרך להתחבר תחילה לשירות SMTP על ידי שימוש באובייקט Session של CDO. קטע התסריט הבא מדגים כיצד להתחבר באמצעות שם המשתמש Administrator, ועם כתובת הדואר האלקטרוני administrator@yourdomain.com :

```
SET mySession = Server.CreateObject("CDONTS.Session")
mySession.LogonSMTP "Administrator", "administrator@yourdomain.com"
```

עליך לקרוא לשיטה LogonSMTP() לאתחול האובייקט Session לפני שתוכל להשתמש במאפיינים ובשיטות האחרים של CDO (מלבד לאובייקט NewMail, כמובן). בסיום השימוש ב-CDO, עליך להתנתק :

```
mySession.Logoff
```

לאחר ההתחברות, עליך להשיג ייחוס (Reference) לתיקיה Inbox. במקרה של שירות SMTP של Microsoft, מקבילה התיקיה Inbox לספריה Drop. ניתן לבצע שלב זה על ידי שימוש במאפיין Inbox של האובייקט Session :

```
SET myInbox = mySession.Inbox
```

לאחר מכן, לאחזור אובייקט המייצג את כל ההודעות ב-Inbox, השתמש במאפיין Messages :

```
SET myMessages = mySession.Messages
```

סוף כל סוף, קיבלנו אובייקט שימושי. האובייקט Messages מכיל אוסף של כל ההודעות בתיקיה Drop. מפני שאובייקט זה הוא אוסף, תוכל להשתמש ב-Item וב-Count כדי לעבור על כל ההודעות באוסף, אחת אחת :

```
FOR k = 1 TO myMessages.Count
    SET myMessages = myMessages.Item(k)
    Response.Write myMessage.Sender & " - "
    Response.Write myMessage.Subject & "<BR>"
NEXT
```

תסריט זה עובר על כל ההודעות שבאובייקט Messages אחת אחרי השנייה, ומציג את שם השולח ואת שורת הנושא של כל אחת מההודעות. שים לב כי כל הודעה באוסף Messages היא אובייקט בפני עצמה. לאובייקט Message מספר מאפיינים שימושיים, כפי שניתן לראות בטבלה 10.1 (הרשימה אינה מלאה, אך היא כוללת את המאפיינים החשובים ביותר).

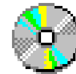
טבלה 10.1 מאפייני האובייקט Message

| מאפיין | תיאור |
| --- | --- |
| Attachments | מחזיר אובייקט Attachment יחיד או אוסף של אובייקטי Attachment, המייצגים את כל הקבצים המצורפים להודעות הדואר האלקטרוני. |
| ContentBase | מחזיר את הכותרת Content-Base, במקרה של הודעת דואר אלקטרוני מסוג MIME HTML. |
| ContentID | מחזיר את הכותרת Content-ID, במקרה של הודעת דואר אלקטרוני מסוג MIME HTML. |
| ContentLocation | מחזיר את הכותרת Content-Location, במקרה של הודעת דואר אלקטרוני מסוג MIME HTML. |
| HTMLText | מחזיר את תכולת גוף הודעת הדואר האלקטרוני בתצורת HTML. |
| Importance | מחזיר את עדיפות ההודעה. ייתכנו שלושה ערכי עדיפויות:<br>1: נמוכה<br>2: רגילה<br>3: גבוהה |
| Recipients | מחזיר אובייקט Recipient יחיד, או אוסף של אובייקטי Recipient, המייצגים את כל נמעני הודעת הדואר האלקטרוני. |
| Sender | מחזיר אובייקט AddressEntry המייצג את שולח הודעת הדואר האלקטרוני. |
| Size | מחזיר את גודלה המשוער של ההודעה בבתים, כולל גודלם של הקבצים המצורפים. |
| Subject | מחזיר את נושא הודעת הדואר האלקטרוני. זהו מאפיין ברירת המחדל של האובייקט Message. |
| Text | מחזיר את תכולת גוף הודעת הדואר האלקטרוני בתצורת טקסט פשוט. |
| TimeReceived | מחזיר את תאריך וזמן קבלת ההודעה או, אם נתון זה אינו קיים, את תאריך וזמן משלוח ההודעה. |
| TimeSent | מחזיר את תאריך וזמן משלוח הודעת הדואר האלקטרוני. |

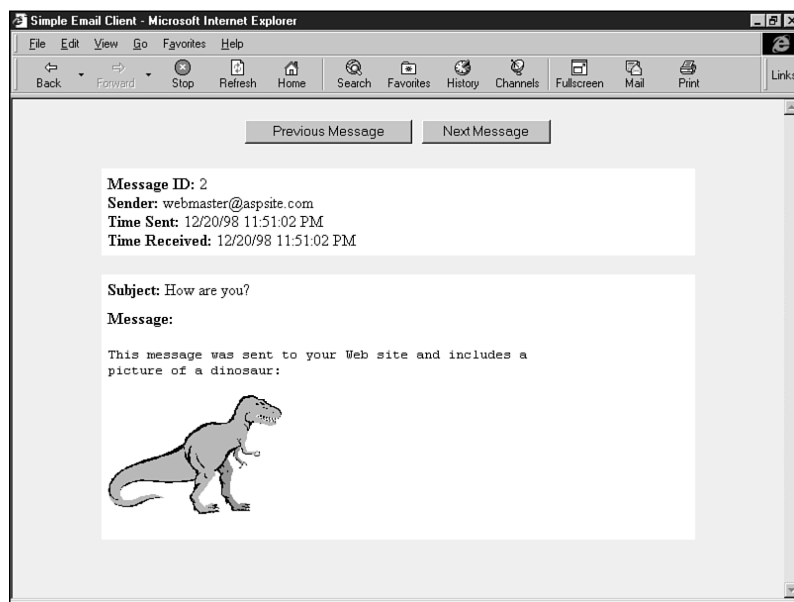
האובייקט Message כולל גם כמה שיטות שימושיות לעבודה עם הודעות בספריה Drop. השיטה ()Delete משמשת למחיקת הודעה בודדת:

myMessage.Delete

במידה ותרגיש הרסני במיוחד, תוכל להפעיל את השיטה ()Delete גם על האוסף Messages כולו. הדבר יסיר את כל ההודעות שבתיקיה Drop לצמיתות.



על ידי שימוש במאפיינים אלה של האובייקט Message, תוכל בקלות ליצור לקוח דואר אלקטרוני פשוט מבוסס אינטרנט (ראה תרשים 10.6). התסריט בתדפיס 10.3 מאפשר לך לאחזר את ההודעות שבספריה Drop, ולצפות בהן אחת לאחר השנייה.




**תרשים 10.6** לקוח דואר אלקטרוני פשוט

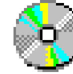
**תדפיס 10.3** לקוח דואר אלקטרוני פשוט (הקובץ EmailClient.asp, נמצא בתקליטור)

```
<%
DIM displayName, emailAddress
DIM mySession, myInbox, myMessages, myMessage
DIM messageIndex
displayName = TRIM( Request( "displayname" ) )
emailAddress = TRIM( Request( "emailAddress" ) )
messageIndex = Request( "messageIndex" )
IF messageIndex = "" THEN messageIndex = 1
messageIndex = cINT( messageIndex )
```

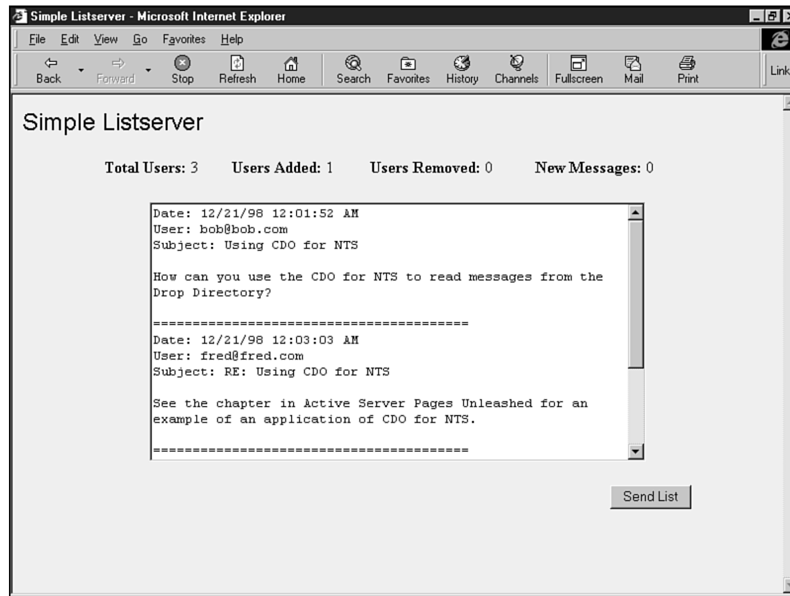
```
IF displayName = "" OR emailAddress = "" THEN
    %>
    <HTML>
    <HEAD><TITLE>Login</TITLE></HEAD>
    <BODY BGCOLOR="#EEEEEE">
    <FORM METHOD="POST" ACTION="emailclient.asp">
    <P>Display Name:  <INPUT NAME="displayName" TYPE="TEXT" SIZE=20>
    <P>Email Address: <INPUT NAME="emailAddress" TYPE="TEXT" SIZE=40>
    <P><INPUT TYPE="SUBMIT" VALUE="Login">
    </FORM>
    </BODY>
    </HTML>
    <%
    Response.End
END IF

SET mySession = Server.CreateObject( "CDONTS.Session" )
mySession.LogonSMTP displayName, emailAddress
SET myInbox = mySession.Inbox
SET myMessages = myInbox.Messages
%>
<HTML>
<HEAD><TITLE>Simple Email Client</TITLE></HEAD>
<BODY BGCOLOR="#EEEEEE">
<CENTER>
<%
IF myMessages.Count = 0 THEN
%>
You do not have any messages.
<%
ELSE
SET myMessage = myMessages.Item( messageIndex )
%>
<% IF myMessages.Count > 0 THEN %>
<TABLE CELLPADDING=4>
<TR>
<% IF messageIndex > 1 THEN %>
    <FORM METHOD="POST" ACTION="emailclient.asp">
    <INPUT NAME="displayName" TYPE="HIDDEN" VALUE="<%=displayName%>">
    <INPUT NAME="emailAddress" TYPE="HIDDEN" VALUE="<%=emailAddress%>">
    <INPUT NAME="messageIndex" TYPE="HIDDEN" VALUE="<%=messageIndex -
1%>">
    <TD>
    <INPUT TYPE="SUBMIT" VALUE="Previous Message">
    </TD>
    </FORM>
<% END IF %>
```

```
<% IF messageIndex < myMessages.Count THEN %>
    <FORM METHOD="POST" ACTION="emailclient.asp">
    <INPUT NAME="displayName" TYPE="HIDDEN"
        ↳ VALUE="<%=displayName%>">
    <INPUT NAME="emailAddress" TYPE="HIDDEN"
        ↳ VALUE="<%=emailAddress%>">
    <INPUT NAME="messageIndex" TYPE="HIDDEN"
        ↳ VALUE="<%=messageIndex + 1%>">
    <TD>
    <INPUT TYPE="SUBMIT" VALUE="Next Message">
    </TD>
    </FORM>
<% END IF %>
</TR>
</TABLE>
<P>
<% END IF %>
<TABLE WIDTH=600 BGCOLOR="#FFFFFF" CELLPADDING=4>
<TR>
    <TD>
    <B>Message ID:</B> <%=messageIndex%>
    <BR><B>Sender:</B> <%=myMessage.Sender%>
    <BR><B>Time Sent:</B> <%=myMessage.TimeSent%>
    <BR><B>Time Received:</B> <%=myMessage.TimeReceived%>
    </TD>
</TR>
</TABLE>
<P>
<TABLE WIDTH=600 BGCOLOR="#FFFFFF" CELLPADDING=4>
<TR>
    <TD><B>Subject:</B> <%=myMessage.Subject%></TD>
</TR>
<TR>
    <TD>
    <B>Message:</B>
    <BR>
    <PRE><%=myMessage.Text%></PRE>
    </TD>
</TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
<%
END IF
SET myMessages = Nothing
mySession.Logoff
SET mySession = Nothing
%>
```

# יצירת Simple List Server



בסעיף זה, שהוא האחרון בפרק, תלמד כיצד ליישם את מה שלמדת בסעיפים הקודמים ליצירת שרת רשימה פשוט (Simple List Server) (ראה תרשים 10.7). באמצעות שרת זה ניתן ליצור קבוצת דיון מבוססת דואר אלקטרוני. (תמצא את הקוד המלא בתקליטור, בקובץ listserv.asp).




**תרשים 10.7** שרת הרשימה הפשוט

לשרת הרשימה הפשוט שלושה תפקודים עיקריים. ראשית, הוא מאפשר למשתמשים חדשים להצטרף לרשימה על ידי משלוח הודעת דואר אלקטרוני שנושאה הוא המילה הבודדת Add. לאחר הצטרפות המשתמש לרשימה, הוא יקבל את מקבץ הודעות הדואר האלקטרוני הנאספות מהמשתמשים האחרים.

שנית, שרת הרשימה הפשוט מאפשר למשתמשים להסיר את עצמם מהרשימה באופן אוטומטי. כל אחד יכול להפסיק את חברותו ברשימה על ידי משלוח הודעת דואר אלקטרוני שנושאה הוא המילה הבודדת Remove.

לבסוף, מקבץ שרת הרשימה את כל הודעות הדואר האלקטרוני הנשלחות אליו להודעת דואר אלקטרוני אחת, אותה הוא מפיץ לכל החברים ברשימה. מקבץ ההודעות נשלח לחברי הרשימה בכל פעם שמפעיל הרשימה לוחץ על Send List (ראה תרשים 10.8).

**תרשים 10.8** משלוח הרשימה לכל חברי שרת הרשימה

# השימוש ב-Simple List Server

כדי להשתמש בשרת הרשימה הפשוט (Simple List Server), יש להעתיק את הקובץ listserver.asp מהתקליטור המצורף לספר אל ספריה הנגישה לשרת האינטרנט (Web Server), דוגמת הספריה wwwroot. לאחר מכן, יש לשנות שני קבועים המופיעים בתחילת קובץ זה.

הראשון הוא הקבוע listServerEmail, המשמש לציון כתובת הדואר האלקטרוני של שרת הרשימה. חברי הרשימה שולחים את הודעות הדואר האלקטרוני שלהם לכתובת זו. עליך לשנות כתובת זו לכתובת הנמצאת ב-domain שלך (לדוגמה, listserver@yourdomain.com).

שני הוא הקבוע userListDir, המספק את נתיב הספריה בה יאוחסנו ההודעות, ורשימת המשתמשים. זו יכולה להיות כל ספריה הנמצאת בשרת (אך עליה באמת להימצא בשרת!). כמו כן, כדי להשתמש בשרת הרשימה, יש להגדיר לשרת האינטרנט הרשאות כתיבה לספריה זו (ראה פרק 8).

בכל פעם שתפעיל את שרת הרשימה הפשוט על ידי טעינת הדף listserver.asp בדפדפן שלך, יעדכן שרת הרשימה באופן אוטומטי את רשימת המשתמשים וההודעות שלו. רשימת חברי הרשימה מאוחסנת בקובץ userlist.txt. מקבץ ההודעות מאוחסן בקובץ messages.txt.

# כיצד פועל Simple List Server

שרת הרשימה הפשוט (Simple List Server) מתפעל את רשימת ההודעות והמשתמשים שלו על ידי מעבר על כל הודעות הדואר האלקטרוני הממוענות אליו בספריה Drop של שרת האינטרנט. לדוגמה, השגרה הבאה משמשת להוספת משתמשים לרשימה:

```
SUB addUsers
    DIM mySession, myInbox, myMessages, myMessage, k
    addUsersCount = 0
    SET mySession = Server.CreateObject( "CDONTS.Session" )
    mySession.LogonSMTP "listserver", listServerEmail
    SET myInbox = mySession.Inbox
    SET myMessages = myInbox.Messages
    IF myMessages.Count > 0 THEN
    FOR k = 1 TO myMessages.Count
        IF UCASE( TRIM( myMessages.Item( k )  ) ) = "ADD" THEN
            SET myMessage = myMessages.Item( k )
            addUser myMessage.Sender
            myMessage.Delete
            addUsersCount = addUsersCount + 1
        END IF
    NEXT
    END IF
    SET myMessages = NOTHING
    mySession.Logoff
    SET mySession = NOTHING
END SUB
```

שגרה זו מתחברת לשרת SMTP ועוברת על כל הודעות הדואר האלקטרוני בזו אחר זו, באמצעות לולאת FOR...NEXT. במקרה של היתקלות בהודעה ששורת הנושא (Subject) שלה היא המילה ADD, מתוסף המשתמש לרשימה על ידי השגרה addUser. כמו כן, נמחקת ההודעה מהספריה Drop, כך שהמשתמש לא יתווסף לרשימה יותר מפעם אחת.

בעת הוספת משתמש לרשימה, נעשה שימוש ברכיב File Access לצירוף כתובת הדואר האלקטרוני של המשתמש לקובץ userlist.txt. הנה השגרה המשמשת להוספת משתמש:

```
SUB addUser( ByVal theUser )
    DIM fs, ts
    IF NOT userExists( theUser ) THEN
        Set fs = Server.CreateObject( "Scripting.FileSystemObject" )
        Set ts = fs.OpenTextFile( userlistDIR & "userlist.txt", 8, TRUE )
        ts.WriteLine UCASE( theUser )
        ts.Close
        SET fs = Nothing
        SET ts = Nothing
    END IF
END SUB
```

שרת הרשימה הפשוט משתמש בשגרות דומות לניהול רשימת ההודעות. הודעות חדשות נמשכות מהספריה Drop ומצורפות לקובץ messages.txt. לאחר הוספת ההודעה לקובץ, היא נמחקת מהספריה Drop באופן אוטומטי.

בעת לחיצה על Send List, נשלחות ההודעות שבקובץ messages.txt לכל אחת מכתובות הדואר האלקטרוני שבקובץ userlist.txt. רשימת המשתמשים נטענת על ידי השגרה הבאה:

```
SUB loadUserList
    totalUsersCount = 0
    DIM fs, tf
    DIM userlistFile
    SET fs = Server.CreateObject( "Scripting.FileSystemObject" )
    IF fs.FileExists( userlistDir & "userlist.txt" ) THEN
    SET tf = fs.OpenTextFile( userlistDir & "userlist.txt" )
        userlistFile = UCASE( tf.ReadAll )
        userList = SPLIT( userlistFile, vbCRLF )
    tf.Close
    SET tf = Nothing
    totalUsersCount = UBOUND( userlist )
    IF totalUsersCount = 0 THEN totalUsersCount = 1
    END IF
    SET fs = Nothing
END SUB
```

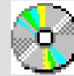
שגרה זו משתמשת ברכיב File Access לטעינת הקובץ userlist.txt. תוכן הקובץ כולו מאוחזר על ידי השיטה ReadAll, ומוקצה למשתנה userlistFile. לאחר מכן, נעשה שימוש בפונקציה ()SPLIT של VBScript ליצירת מערך של כל המשתמשים, על ידי פיצול תוכן המשתנה userlistFile בכל מופע של תו Enter (Carriage Return) והזנת שורה (Line Feed). מערך משתמשים זה הוא התוצאה הסופית של השגרה loadUsersList.

התסריט הבא משמש למשלוח רשימת ההודעות לכל המשתמשים שבמערך userlist:

```
IF totalUsersCount > 0 THEN
FOR i = 0 TO UBOUND( userlist )
IF userlist( i ) <> "" THEN
 %>
<BR>Sending list to user <%=i + 1%> at email addess <%=userlist( i )%>...
<%
sendMail listServerEmail, userlist( i ), listServerSubject, messages
END IF
NEXT
```

תסריט זה משתמש בלולאה מסוג FOR...NEXT למעבר על כל המשתמשים. השגרה sendMail (נידונה קודם לכן בפרק זה) שולחת את ההודעות.

**הערה:**



הפונקציה ()SPLIT של VBScript יודעת לקבל מחרוזת טקסט ולהפוך אותה למערך על פי תו הנתון כפרמטר האומר איפה לחתוך את המחרוזת (ראה נספח C).

# סיכום

בפרק זה, למדת את כל סודות השימוש בדואר אלקטרוני באתר האינטרנט שלך. בחלק הראשון, למדת כיצד לקבוע את תצורת שירות SMTP של Microsoft. בחלקים השני והשלישי, למדת כיצד להשתמש ב-CDO for NTS לצורך אינטראקציה עם שירות SMTP, למטרת משלוח הודעות דואר אלקטרוני מתוך תסריטי דפי ASP. לבסוף, נידונה דוגמה ליישום העושה שימוש בשירות SMTP וב-CDO for NTS. למדת כיצד ליצור שרת רשימה פשוט.

## פרק 11

# עבודה עם ממשקי שירות של Active Directory

**פרק זה מיועד למתקדמים.**

**בפרק זה:**

- מהו Directory?
- מהו Directory Service?
- מהו Active Directory?
- מבט כללי על מודל האובייקטים של ADSI
- Property Caching
- בחינת Domain
- תפעול משתמשים וקבוצות
- שליטה בשירותי NT
- שליטה במדפסות
- ניהול שרת Exchange
- ADSI ו-Site Server
- שימוש ב-ADO עם ADSI
- שיקולי אבטחה
- שימוש ב-ADSI עם IIS Metabase

ADSI הם ראשי תיבות של Active Directory Services Interface (ממשק לשירותי Active Directory). מה זה אומר, אם כך? מהשם, ניתן להניח כי ADSI הוא כנראה ממשק. ואכן, ADSI הוא אוסף של אובייקטי COM המספקים שיטות (Methods) ומאפיינים (Properties) באמצעותם יוכל יישום (Application) לקבל שירותים (Services). מהשם, ניתן לשער כי ADSI קשור איכשהו ל-Active Directory. ואכן, ניתן להשתמש ב-ADSI כממשק ל-Active Directory. מאידך, ניתן גם להסיק מהשם כי ADSI הוא ממשק ל-Active Directory Service.

בפרק זה נבחן את ADSI ממספר זוויות. נסתכל על הדגם הכללי (גנרי) של Directory Service. נסתכל על מה שמתיימר ה-Active Directory של Microsoft להיות. אבל, את מרבית הזמן נבלה בהתבוננות בממשקים (אובייקטים) המרכיבים את ADSI.

פרק זה לא יוכל לספק פרטים ממצים ודוגמאות בנוגע לכל ההיבטים של ADSI. הנושא רחב מדי לפרק אחד. מטרת הפרק היא להבהיר לקורא מהו ADSI, וכן לספק מספר דוגמאות ברורות, פשוטות ושימושיות לאופן בו ניתן להשתמש ב-ADSI כמנוף לקוד ה-ASP.

כמו כן, הדוגמאות בפרק זה נכתבו כך שיראו את הקלות שבשימוש ב-ADSI. הדוגמאות המופיעות כאן אינן כוללות לוגיקת איתור ולכידת שגיאות, וזאת בכוונה תחילה. נמנעתי גם מקוד שמטרתו לקשט את הפלט. במקום זאת, הדוגמאות הן קצרות וזריזות, ומדגימות את הממשק התכנותי של ADSI.

הערה חשובה בנוגע לדוגמאות שבפרק זה. הדוגמאות פותחו באמצעות ADSI גירסה 2.0, ודורשות כי בשרת ה-IIS בו יופעלו הדוגמאות יותקן גם ADSI SDK (ADSI Software Development Kit). ניתן להוריד את ה-SDK עבור ADSI מהאתר של Microsoft. בנוסף, חלק מהדוגמאות דורשות התקנה של מוצרי Microsoft אחרים, כגון Exchange Server גירסה 3.5 ו-Site Server גירסה 3.0, כדי שיוכלו לפעול כראוי. והערה אחרונה – הפכתי את הדוגמאות לכלליות, בכך שהחלפתי שמות שרתים מפורשים במחרוזת "server". יש להחליף מחרוזת זו במחרוזת המייצגת את השרת עליו תופעל הדוגמה.

# מהו Directory?

החיים מלאים בדוגמאות למדריכים (Directories). ספר הטלפונים, למשל. ספר הטלפונים מספק מאגר מרכזי לאחסון מידע אודות אנשים ו/או שירותים. ספר הטלפון הוא מדריך הכולל שמות, כתובות ומספרי טלפון של אנשים. ניתן למצוא רשומה מסוימת במדריך בקלות, כיון שהרשומות ממוינות על פי שם משפחה. לגבי חלק מהרשומות, ניתן ללקט, מלבד למספר הטלפון, גם מידע נוסף. האם האדם הוא זכר או נקבה? האם הוא דוקטור? האם מספר אנשים משויכים לאותה הכתובת?

דפי זהב הוא מדריך מסוג אחר. מדריך זה מאורגן על פי סוג השירות. כאן ניתן למצוא, בנוסף לשם, הכתובת ומספר הטלפון המשויכים לכל רשומה, המון מידע נוסף (תכונות) לגבי השירות המוצע.

בדרך כלל, המדריכים מאורגנים לפי פרמטר אחד לפחות, כדי שהחיפוש בהם יהיה יעיל. ספר הטלפון מאורגן למציאה של אנשים על פי שם משפחה. דפי הזהב דורשים מהמשתמש לדעת את סוג השירות או המוצר המבוקש.

# מהו Directory Service?

בעולם המחשבים, נעשה שימוש ב-Directory Service למעקב אחר ישויות החשובות לתחומי מערכות המידע (IS, Information Systems) או טכנולוגיית המידע (Information Technology, IT). ה-Directoty Service עוקב אחר פרטים כגון איזה מחשב מחובר לאיזה Domain, אילו מדפסות קיימות והיכן הן נמצאות, מי הם משתמשי המערכת, ומהן זכויות הגישה (Access Rights) המוענקות לכל משתמש. הרחבה של מודל זה מספקת גם מעקב אחר מידע משני בנוגע למשתמשים, כדוגמת כתובות דואר אלקטרוני, מספרי טלפון, וכתובות.

כשארגון מתרחב לסדר גודל של מעבר לרשת מקומית (LAN) יחידה, נעשה קשה יותר ויותר לנהל ולעקוב אחר כל המשאבים (כולל אנשים) המקושרים עם הארגון. תאר לעצמך ארגון בינלאומי, המעונין במדריך יחיד שיכלול את כל עובדי הארגון, כך שניתן יהיה להפנות דואר אלקטרוני בקלות לכל אחד מחברי הארגון או לכולם ביחד.

תעשיית המחשבים פיתחה מספר פתרונות לבעיות אלו. חברת נובל (Novell) פיתחו את Bindary עוד בימי NetWare 3.x. Bindary סיפקה Directory SErvices לסביבת LAN. Bindary אפשרה מתן שמות למשאבי רשת (Networking Resources), כגון מדפסות ומערכות קבצים.

Novell הגדילה את ההיקף של Bindary כשהציגה את ה-Directory Services כחלק מ-NetWare 4.0. ה-Directory Services המשופרים תוכננו לתמוך בארגונים גדולים מאד. הם התאימו לרשתות מרחביות (WAN) המורכבות ממספר רשתות מקומיות (LANs).

פתרון נוסף לניהול משאבי הרשת בארגונים, צמח בדמות תקן X.500. תקן X.500 פותח על ידי ה-Open System Foundation כפתרון למודל OSI. התקן סיפק מפרט עמיד ביותר ל-Directory Services ברמת הארגון. לרוע המזל, מפרט X.500 הוא פתוח עד כדי כך שהוא דורש מיישומים שלו לתמוך בכמות רבה ביותר של תצורות מידע (Data Formats), טרנספורמציות, פרוטוקולי תקשורת וכן הלאה.

לארגונים רבים, היה התקן X.500 הרבה מדי. הצורך ב-Directory Services פשוט הוביל לפיתוח המפרט **L**ightweight **D**irectory **A**ccess **P**rotocol ‏(LDAP). ‏IETF פיתחו חלופה למפרט LDAP ‏(RFC 1777), שהגדירה תת מערכת ממפרט X.500. ‏LDAP תוכנן בסביבת TCP/IP וסיפק תמיכה רק לממשקים הבסיסיים הדרושים ל-Directory Services שימושי.

Microsoft אימצה את מפרט LDAP, והשתמשה בו כדי לספק שירותי חברות (Membership) והתאמה אישית (Personalization) ל-Site Server 3.0, ערכת הכלים של Microsoft המיועדת לפיתוח אתרי אינטרנט הכוללים רשתות אינטראנט ואתרי מסחר אלקטרוני. Microsoft אימצה את LDAP גם כמחסן הכתובות ואנשי הקשר של שרתי Exchange. פרק זה יכסה את LDAP ביתר פירוט בהמשך.

Microsoft מאמצת אף יותר את ה-Directory Services במערכת ההפעלה Windows 2000, שהיא צעד נוסף לקראת איחוד והחלפת Windows 98 ו-Windows NT. ה-Directory Service הוא רכיב מרכזי ובסיסי ב-Windows 2000, והוא מחליף את המודל הקיים, המבוסס על מודל PDC/BDC/Domain.

באופן כללי, Directory Services הוא מחסן מרכזי של מידע לגבי האובייקטים המתוחזקים בארגון או ברשת. האובייקטים הספציפיים והמידע לגביהם ניתנים, בדרך כלל, להרחבה. Directory Service מייצג Persistent Store המאחד סביבות הטרוגניות.

# מהו Active Directory?

מערכת האבטחה של Windows NT מבוססת על Domain Model ועל מנהל אבטחת גישה (Security Access Manager, SAM), המנהלים מעקב אחר משתמשים וקבוצות. למרבה הצער, הגבול המעשי ב-Domain מבוסס Windows NT הוא 20,000 משתמשים. ניתן להגדיל מספר זה על ידי השימוש במספר domains המחוברים זה לזה באמצעות יחסי אמון (Trust Relationship), אך אלה יכולים להסתבר כמגושמים לניהול. Windows NT היא פלטפורמה מצוינת לבניית סביבות LAN, אך חסרה לה האפשרות לתמיכה בארגונים גדולים מאוד (אם כי אני משוכנע ש-Microsoft לא יסכימו איתי בנקודה זו).

עוד פגם או מכשול שהפריע לאימוץ Windows NT כמערכת הפעלה לארגונים היה יתרונה של Novell וה-Directory Service שלה.

Microsoft מנסה לשנות זאת על ידי החלפת הארכיטקטורה המבוססת על PDC, ‏BDC ו-SAM, בתשתית חדשה מבוססת LDAP, המכונה Active Directory (מוכר גם בקיצור כ-AD). ‏AD פותר את נושאי יכולת הגידול (Scalability) המטרידים של Windows NT, ומאמץ לגמרי את פתרון Directory Service של Novell. ‏Windows 2000 מאפשרת למוצרי Microsoft להשתלט גם על מטלות הניהול היומיומיות של רשתות Novell קיימות. מובן, כי Microsoft מקווה כי כשיתעורר צורך בהתקנת שרת חדש, יבחרו הצרכנים בפלטפורמת Windows 2000.

אחד מיעדיו העיקריים של ה-AD הוא לאפשר עבודה עם ארגונים גדולים במיוחד, המורכבים ממיליוני אובייקטים המשתרעים על סביבות שהן גדולות, ולעיתים קרובות גם מפוזרות. AD מבטיח את היכולת המבוקשת לאפשר תמיכה בשם משתמש וסיסמה יחידים על פני מספר רשתות LAN שונות, הכוללות סביבות מערכת הפעלה שונות. כניסות יחידות (Single Logons) לשרתי קבצים, מערכות דואר אלקטרוני ורשתות חיצוניות (Extranets) יפחיתו את הוצאות הבעלות (Total Cost of Ownership ,TCO) המיוחסות לסביבות מרושתות של ארגונים גדולים (Enterprise-Wide networking environments).

Active Directory (AD) מושתת על מושגים כגון יערות (Forests), עצים (Trees) ו-Domains. יער הוא אוסף של אחד או יותר עצים המשותפים ביחסי אמון (Trust Relationship). כל עץ הוא אוסף של אחד או יותר domains החולקים סכימה כללית. כל domain מחולק לאחת או יותר יחידות ארגוניות (Organizational Units ,OU), שהן ישויות לוגיות המנוהלות עצמאית.

ראשו של כל עץ מזוהה בדרך כלל על ידי שם ארגוני (Oragnizational Name), (O), המשמש כאוסף של יחידות ארגוניות (Oraganizational Units), (OU) אחרות. יחידה ארגונית (OU) היא יישות עצמאית המתפעלת משתמשים, קבוצות, מנהלים (Administrators) ומשאבים משל עצמה. לעיתים קרובות, מנוהלות יחידות ארגוניות באופן עצמי (Self Managed). לדוגמה, יחידה ארגונית יכולה להיות על בסיס גאוגרפי: תל אביב, ירושלים, אילת וחיפה. כל OU יכולה לכלול קבוצת מנהלים (Administrators Group), המתפעלת את האובייקטים החברים ב-OU.

AD הוא אחד ממרחבי שמות (Namespaces) רבים הנתמכים על ידי ADSI. מרחבי שמות אחרים כוללים את מרחב הכתובות הנוכחי של Windows NT, NetWare Bindary, NetWare Directory Service, LDAP, ו-NFS. מרחב השמות של AD מורכב משמות ייחודיים (Distinguished Names, DN). DN מתחיל בפרטי ומתקדם דרך מרחב השמות עד לכללי. שמות AD כוללים מילות מפתח הלקוחות ממודל X.500 (ו-LDAP). טבלה 1.1 מתארת את היישויות היוצרות את שמות ה-AD:

**טבלה 11.1** יישויות Active Directory

| המפתח | תיאור | תיאור באנגלית |
|---|---|---|
| O | ארגון | Organization |
| DC | רכיב domain | Domain Component |
| OU | יחידה ארגונית | Organizational Unit |
| C | שם מדינה | Country Name |
| ST | מדינה/מחוז | State/Province |
| L | אזוריות | Locality |
| CN | שם כללי | Common Name |

ל-LDAP ול-AD מספר מוסכמות משותפות בנוגע לשמות אובייקטים, לדוגמה ארגון (O), יחידה ארגונית (OU) ו-רכיב domain (DC). למעשה, AD מבוסס על LDAP ומייצג מקרה נוסף בו Microsoft אימצה תקן ושיפרה אותו. AD הוא מיקוד של מודל הסכימה כללית (גנרית), הנתמך על ידי LDAP.

בתיעוד ADSI המסופק על ידי Microsoft, נטען כי ADSI הוא "X.500 המותאם במיוחד ל-LDAP ול-Novell Directory Services (NDS)". אין ספק כי ארגון מרחבי השמות של שני Directory Services נפוצים אלה, מאומץ ומקודם הן על ידי AD והן על ידי ADSI.

במצב כיום, יש לכל חבילת Directory Services תחביר מרחב שמות משלה. AD מבטיח לאחד את עולם מרחבי השמות, כך שבבואם לתחזק ארגון המורכב ממערכות שונות יצטרכו מנהלי או מתכנתי IT להתמודד עם תחביר מרחבי שמות אחד בלבד. AD יקל על העבודה עם NFS, NetWare, Windows NT וסביבות רשת שונות נוספות בתוך ארגון יחיד (עם כל האיחודים והרכישות השכיחים כל כך בימינו, אוספים מאין אלה אינם נדירים כלל וכלל).

# מבט כללי על מודל האובייקטים של ADSI

ADSI מיישם מודל אובייקטים התומך בייצוג Directory Services קיימים רבים. נכון להיום קיימים ספקי שירות ADSI ל-Windows NT 4.0, Windows 2000, NetWare 3, NetWare 4, LDAP ו-NFS. ADSI מספק סביבה גמישה, בה ניתן לעצב את מרחבי השמות הקיימים. ADSI מאפשר גם יצירה והרחבה של סכימה, דבר המאפשר יכולות הרחבה. לדוגמה, אם יחידה ארגונית מחליטה כי קיימת חשיבות למעקב אחר מספר תעודת הזהות או כרטיס האשראי של המשתמש, יש ביכולתה להוסיף שדות אלה לסכימה של האובייקט User.

אם תשקיע זמן בהורדת תיעוד ADSI מרשת האינטרנט ועיון בו, תמצא מידע רב לגבי גישה ל-ADSI באמצעות C++, Java, VB וכן הלאה. כמתכנת ASP, תצטרך להתמקד בממשקים המתחילים ב-IADsXXX בלבד. אלה הם ממשקי האוטומציה של COM, הנגישים דרך ASP.

ADSI מספק שלושה אובייקטי ממשק הנמצאים בכל מקום בהיררכיית האובייקטים של ADSI. אובייקטים אלה הם IADs, IADsContainer ו-IADsPropertyList.

הממשק IADs הוא בסיסי לכל אובייקטי ADSI, ואחריותו היא זיהוי האובייקטים. ממשק זה דומה למחלקת בסיס, שכל שאר מחלקות ADSI יורשות ממנה. ממשק IADs מספק שיטות גישה למאפיינים. המאפיינים הם המבדילים בין אובייקט ADSI אחד לאחר. הממשק IADs מספק גם גישה לסכימה המתארת את ה-metadata של אובייקט ADSI. כל אובייקטי העלה (Leaf Objects) בעץ ה-Directory של ADSI (ADSI Directory Tree) נגזרים מ-IADs. טבלה 11.2 מתארת את המאפיינים והשיטות המשויכים לממשק IADs.

**טבלה 11.2** שיטות ומאפיינים ב-IADs

| שיטה/מאפיין | תיאור |
| --- | --- |
| Name | שמו היחסי של האובייקט |
| Class | שם המחלקה של סכימת האובייקט |
| GUID | המזהה הגלובלי הייחודי של האובייקט |
| AdsPath | נתיב המזהה את האובייקט באופן ייחודי במרחב השמות |
| Parent | נתיב AdsPath להורה האובייקט |
| Schema | נתיב AdsPath של הסכימה המשויכת לצומת זו |
| SetInfo() | שמירת שינויים שנעשו באובייקט ובתלויים בו |
| GetInfo() | טעינה מחדש של המאפיינים המשויכים עם האובייקט ממחסן הנתונים הבסיסי |
| Get() | השגת ערך עבור מאפיין מסוים |
| Put() | קביעת ערך עבור מאפיין מסוים |
| GetEx() | השגת ערך עבור מאפיין רב-ערכים מסוים |
| PutEx() | קביעת ערך עבור מאפיין רב-ערכים מסוים |
| GetInfoEx() | טעינת המאפיינים המשויכים עם האובייקט ממחסן הנתונים הבסיסי |

IADsContainer הוא הממשק הנדרש לאובייקטי ADSI היוצרים, מוחקים, מעתיקים, מעבירים ומונים (Enumerate) אובייקטי ADSI אחרים. אובייקטי IADsContainer משמשים ליצירת ענפים בעץ ה-Directory. ממשק IADsContainer מרחיב את כל המאפיינים והשיטות של IADs, ומוסיף שיטות ומאפיינים להרחבת הפונקציונאליות הבסיסית. טבלה 11.3 מתארת את המאפיינים והשיטות הנוספים השותפים בממשק IADsContainer.

**טבלה 11.3** שיטות ומאפיינים ב-IADsContainer

| שיטה/מאפיין | תיאור |
|---|---|
| Count | מספר האובייקטים במכולה |
| Filter | המסנן שבשימוש בעת מניית של תכולת המכולה |
| Hints | המאפיינים שיוטענו (עצה בלבד) |
| GetObject() | השגת ממשק לאובייקט הנקוב |
| Create() | יצירת האובייקט המצוין |
| Delete() | מחיקת האובייקט המצוין |
| CopyHere() | העתקת האובייקט המצוין למיקום אחר במרחב השמות |
| MoveHere() | העברת האובייקט המצוין למיקום אחר במרחב השמות |

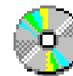
IADsPropertyList הוא ממשק המאפשר גישה למאפיינים של אובייקט. טבלה 11.4 מתארת את ממשק זה.

**טבלה 11.4** שיטות ומאפיינים ב-IADsPropertyList

| שיטה/מאפיין | תיאור |
|---|---|
| PropertyCount | מספר המאפיינים ברשימה |
| Next() | השגת המאפיין הבא ברשימה |
| Skip() | התקדם n מאפיינים ברשימה |
| Reset() | התחל מחדש, בתחילת הרשימה |
| Add() | הוספת מאפיין לרשימה |
| Remove() | הסרת מאפיין מהרשימה |
| Item() | השגת המאפיין הנקוב |
| GetPropertyItem() | השגת ערך המאפיין הנקוב |
| PutPropertyItem() | קביעת ערך המאפיין הנקוב |
| PurgePropertyList() | מחיקת כל המאפיינים מהרשימה |

ADSI משתמש בממשקי הבסיס כדי ליישם התמקדויות רבות, כפי שתיווכח בהמשך פרק זה.

ADSI משתמש בממשקים אלה לבניית ייצוג היררכי של כל מרחבי השמות הזמינים. למעשה, שורש העץ הוא אובייקט IADsCollection בשם IADsNamespaces. IADsNamespaces מאפשר מעבר על כל מרחבי השמות הזמינים בזה אחר זה (כל מרחב שמות נתמך על ידי ספק שירות). השורש, אובייקט מרחב השמות הנמצא ברמה הגבוהה ביותר של ADSI, מקבל את השם :ADS. כדי לעבור על כל מרחבי השמות בזה אחר זה יש לאגד (To Bind) לשורש, ולאחר מכן לעבור על כל מרחבי השמות החברים (Member Namespaces) בזה אחר זה.



הדוגמה בתדפיס 11.1 מציגה יצירת רשימה של כל מרחבי השמות הזמינים ברשת שלך. ניתן למצוא את קוד המקור בתקליטור המצורף לספר זה, בקובץ Namespaces.asp.

**תדפיס 11.1** יצירת רשימה של מרחבי שמות

```
<%
On Error Resume Next

Dim oNameSpaces
Dim oNameSpace

Set oNameSpaces = GetObject("ADS://")
if Err = 0 Then
    Response.Write oNameSpaces.name & "<BR>"

    For each oNameSpace in oNameSpaces
        Response.Write "   " & oNameSpace.name & "<BR>"
    Next
Else
   Response.Write "ADSI error - " & Hex( Err )
End if

%>
```

ADSI מיישם היררכיית סכימה יחד עם היררכיית מרחבי השמות. כל מרחב שמות מספק תיאור של עצמו דרך סכימה המצורפת לשורש מרחב השמות. הסכימה מורכבת מממשקי IADsClass, IADsProperty ו-IADsSyntax. הסכימה עצמה היא אובייקט IADsCollection, המאפשר מעבר בזה אחר זה על כל אובייקטי המחלקה, המאפיינים והתחביר המגדירים את מרחב השמות (ייתכן ולענפים שונים של מרחב השמות, דוגמת יחידות ארגוניות, OUs, יהיו סכימות שונות שיבטלו את סכימת מרחב השמות המוגדרת בשורש עץ מרחבי השמות).

אחת הדרישות מ-Directory Service, ומכאן גם מ-ADSI, היא כי לכל יישות במרחב השמות המנוהל על ידי Directory Service יהיה זיהוי חד ערכי. ADSI משתמש בנתיב AdsPath לזיהוי חד ערכי של כל צומת בספריה. נתיב AdsPath בנוי משני רכיבים - ProgID של ספק מרחב השמות, ומחרוזת ייחודית לספק המזהה את הצומת באופן חד ערכי בתוך מרחב השמות. WinNT ו-LDAP הם דוגמאות לחלקי ה-ProgID של AdsPath.

ה-ProgID מופרד מהחלק מוגדר-מרחב-השמות על ידי //:. לפניך מספר דוגמאות לנתיבי AdsPath:

WinNT://Stellcom1/Uranus/HP Color Printer 3rd floor

LDAP://O=Stellcom, OU=Engineering, CN=HPLJ5

LDAP://CN=HPLJ5, OU=Engineering, O=Stellcom

שים לב כי שמות LDAP ניתנים לעיצוב הן מהפרטי לכללי (Specific To General) והן מהכללי לפרטי (General To Specific). הדרך המועדפת ליצירת שמות LDAP ייחודיים (LDAP Distinguished Names - DN) היא מהפרטי לכללי.

ADSI מספקת מנגנונים נוספים לסיווגם של האובייקטים, המופשטים ומנוהלים בסביבת Directory Service. אובייקטים אלה כוללים את הממשקים IADsUser, IADsDomain ו-IADsOU. ממשקים אלה מסופקים במיוחד למטרת תמיכה במרחבי השמות של LDAP ושל AD.

כמו כן, מסופקים ממשקים לתמיכה באבטחה (Security) ובאימות משתמשים (User Authentication). ממשקים אלה הם IADsAccessControlEntry, IADsAccessControlList, IADsSecurityDescriptor ו-IADsOpenDSObject. בממשקים אלה נתמקד בשלב מאוחר יותר של פרק זה.

# Property cashing

אחת התכונות החשובות של ADSI, הן מטעמי יעילות והן מטעמי תפעול, היא הכנסת המאפיינים למטמון (Property Caching) המסופקת על ידי הממשק IADs. לפני מספר פסקאות למדנו כי לממשק IADs מספר שיטות שנועדו לאחזור וקביעה של ערכי מאפיינים בממשקי ADSI שונים. טבלה 11.5 מפרטת שיטות אלו פעם נוספת, למען נוחות העיון.

**טבלה 11.5** שיטות IADs הקשורות להטמנת מאפיינים

| שיטה | תיאור |
|---|---|
| SetInfo() | שמירת מאפייני המטמון |
| GetInfo() | טוען את מאפיין המטמון מה-Persisted Store |
| Get() | מאחזר מאפיין מסוים מהמטמון |
| Put() | מאחסן מאפיין מסוים למטמון |
| GetEx() | טוען מאפיין רב-ערכי מהמטמון |
| PutEx() | מאחסן מאפיין רב-ערכי במטמון |
| GetInfoEx() | טוען את מטמון המאפיין מה-Persisted store |

ADSI מתפעל הכנסת מאפיינים למטמון, דבר המשחרר את שגרות התוכנה הנמוכות מהצורך לגשת ל-Persisted Data Store בכל קריאה או כתיבה של מאפיין. לוגיקת הכנסה למטמון זו מביאה לשיפור ניכר בביצועים, במיוחד בעת שינוי של מספר מאפיינים המשויכים לאותו האובייקט. דוגמה טובה למצב כזה היא הגדרה של משתמש חדש וקביעת התצורה עבורו. לרוע המזל, בעת כתיבת קוד ADSI יש לקחת בחשבון גם את הכנסת המאפיינים למטמון, וזאת כדי למנוע כשלים לוגיים (Logic Errors).

אופן הכנסת מאפייני ADSI למטמון הוא פשוט למדי. המטמון מתמלא בכל קריאה לשיטות GetInfo() או GetInfoEx(). במידה ואף אחת מהשיטות הללו לא נקראה, ומתבצעת קריאה ל-Get() או ל-GetEx(), מתמלא המטמון באופן אוטומטי. טעינה מפורשת זו של המטמון תתבצע רק בעת הקריאה הראשונה ל-Get() או ל-GetEx().

המטמון הופך למתמיד (Persisted) בעת קריאה ל-SetInfo(). השיטות Set() ו-SetEx() כותבות מידע למטמון בלבד. המטמון לא יהיה Persisted אלא לאחר קריאה ל-SetInfo().

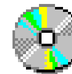
שים לב – בעת קביעה ואחזור של ערכי מאפיינים, יש לוודא כי התבצעה קריאה ל-SetInfo(), כדי להבטיח את שמירת המטמון קודם לקריאת הערכים ממנו. קריאה ל-GetInfo() תמלא את המטמון (Cache) מחדש בערכים מה-Persisted Store. כל ערך ששונה על ידי Put() או PutEx() ולא קובע על ידי SetInfo() יאבד בעת קריאה ל-GetInfo() או GetInfoEx().

תכונה (או תופעת לוואי) מעניינת נוספת של ההכנסה למטמון (Caching) היא ההתנהגות המתגלית בעת נסיון של מספר יישומים שונים לשנות ערכים באותה קבוצת מאפיינים. אין מנגנון אוטומטי שיודיע ליישום על שינויים שהתבצעו ב-Persisted Store לאחר טעינת המטמון. כדי לראות את השינויים שבוצעו על ידי יישומים אחרים, ייאלץ היישום לקרוא ל-GetInfo() פעם נוספת. נוצר מצב של תחרות, בו ינצח היישום שיקרא אחרון ל-SetInfo(). במצב כזה, קל מאד ליישום לדרוס בטעות שינויים שבוצעו על ידי יישום אחר (בעיה זו ניתנת בדרך כלל לפתרון, באמצעות נעילת רשומות (Record Locking), אך ADSI אינו תומך בנעילת רשומות).

עניין אחרון שיש להיות ערים אליו בעת קביעת ערכי המאפיינים, הוא העובדה שאין כל ערובה לכך שכל המאפיינים אכן יישמרו לאחר הקריאה ל-SetInfo() (השיטה SetInfo() אינה שותפה אמינה לתנועות (transactions). בדרך כלל, כל המאפיינים נשמרים בקריאה ל-SetInfo(), אולם תמיד קיימת סבירות נמוכה לתקלה שתתרחש בדיוק באמצע שלב שמירת הנתונים. במקרה כזה, ייתכן ומספר ערכים של המאפיינים יאבדו.

חלק ממגרעות אלו של ADSI ישופרו בגרסאות הבאות. על אף עניינים אלה, ADSI הוא כלי חזק שכדאי (וצריך) לאמץ.

הכנסת המאפיינים למטמון משפיעה גם על פעולתה של IADsContainer. אובייקטים המוספים למכולה (Container) זו אינם נעשים Persisted, עד לקריאה ל-SetInfo(). לעומת זאת, מחיקת אובייקטים משתקפת ב-Persisted store באופן מיידי (בדרך כלל ה-Persisted Data Store הוא שרת SQL, אולם אין כל מניעה שיהיה Access, מערכת מסד נתונים אחרת, או כל סוג אחר של Persisted Data Store הנתמך על ידי ספק שירות מרחב השמות, Namespace Service Provider).



הקוד שבתדפיס 11.2 מדגים את אופן הגישה למאפיינים ב-ADSI. ניתן למצוא את הקוד גם בתקליטור המצורף, בקובץ Properties.asp.

**תדפיס 11.2** גישה למאפיינים דרך ADSI

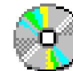
```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<%

Dim oContainer
Dim oRecipient
Dim oProperty
Dim idx

Set oContainer =
    ➥GetObject("LDAP://server/o=organization/ou=site/cn=Recipients")
oContainer.GetInfo()   'load the property cache

For Each oRecipient in oContainer
    Response.Write oRecipient.Name & "<BR>"

    For idx = 1 to oRecipient.PropertyCount
        Set oProperty = oRecipient.Next
        Response.Write "   "
        Response.Write oProperty.Name
        Response.Write " - "
        Response.Write oRecipient.Get(oProperty.Name)
        Response.Write "<BR>"
    Next
Next

%>

</BODY>
</HTML>
```

דוגמה זו יוצרת רשימה של כל הנמענים השייכים לשרת Exchange. ברשימה מוצג שמו של הנמען, ולאחריו המאפיינים המשויכים למען זה (מאפייני הנמענים כוללים מידע כגון מספרי טלפון, מספרי פקס, כתובות וכן הלאה).

הפעלת הדוגמה הקודמת דרשה גישה לשרת Exchange (בדוגמה זו נעשה שימוש בשרת Testbox). יש להפעיל את התסריט כשאתה מחובר לשרת Exchange כבעל זכויות של מנהל מערכת (System Administrator), כדי להבטיח את קיומן של הרשאות גישה מספקות.

# בחינת Domain

ADSI מספק גישה ל- Domain Object, התומך בשליטה על מדיניות ברמת המערכת (System Level Policies). אחת המעלות של ADSI היא היכולת להשתמש בממשקי סכימה (Schema Interfaces) לקביעת פרטי האובייקט. קוד ASP בתדפיס 11.3 מציג שאילתות לסכימת מרחב השמות של WinNT, כדי לגלות את סוגי האובייקטים הכלולים בה, וכן את מאפייני החובה והרשות המשויכים עם כל אובייקט.



ניתן למצוא את הקוד הבא גם בתקליטור המצורף, בקובץ WhoAmI.asp.

**תדפיס 11.3** הצגת שאילתה לסכימת ADSI

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" CONTENT="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<%

Dim oMe
Dim oMySchema
Dim oProperty
Dim oMember

On Error Resume Next

' bind to a domain object
Set oMe = GetObject( "WinNT://stellcom1" )

' properties of the domain object
Response.Write "Domain's definition -<BR>"
Response.Write "Name = " & oMe.name & "<BR>"
Response.Write "ADsPath = " & oMe.ADsPath & "<BR>"
Response.Write "Class = " & oMe.Class & "<BR>"
Response.Write "GUID = " & oMe.GUID & "<BR>"
Response.Write "Parent = " & oMe.Parent & "<BR>"
Response.Write "Schema = " & oMe.Schema & "<BR>"
```

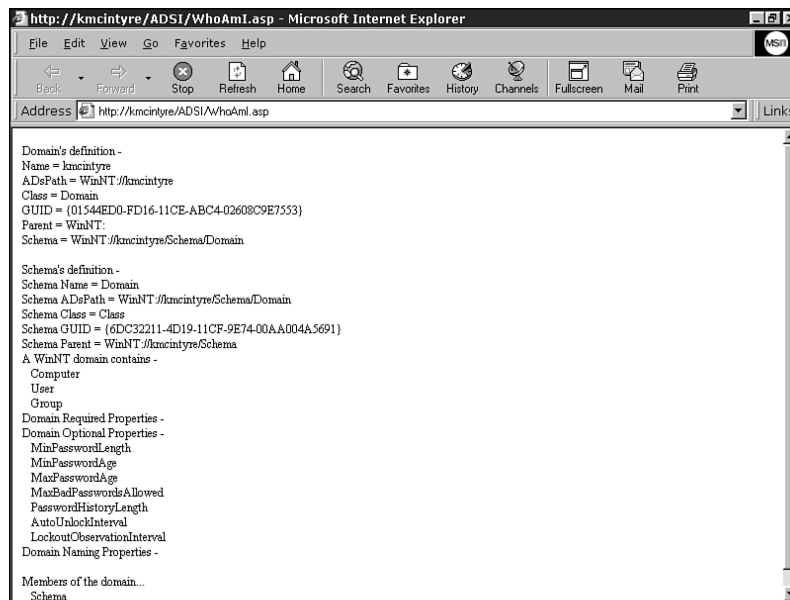
```
' the domain's schema
Response.Write "<BR>Schema's definition -<BR>"
Set oMySchema = GetObject( oMe.Schema )
Response.Write "Schema Name = " & oMySchema.Name & "<BR>"
Response.Write "Schema ADsPath = " & oMySchema.ADsPath & "<BR>"
Response.Write "Schema Class = " & oMySchema.Class & "<BR>"
Response.Write "Schema GUID = " & oMySchema.GUID & "<BR>"
Response.Write "Schema Parent = " & oMySchema.Parent & "<BR>"
' the schema is a container
if oMySchema.Container Then
    Response.Write "A WinNT domain contains -<BR>"
    Response.Write "    derived from " & oMySchema.DerivedFrom
    For Each oProperty in oMySchema.Containment
        Response.Write "   " & oProperty & "<BR>"
    Next

    Response.Write "Domain Required Properties -<BR>"
    For Each oProperty in oMySchema.MandatoryProperties
        Response.Write "   " & oProperty & "<BR>"
    Next
    Response.Write "Domain Optional Properties -<BR>"
    For Each oProperty in oMySchema.OptionalProperties
        Response.Write "   " & oProperty & "<BR>"
    Next
    Response.Write "Domain Naming Properties -<BR>"
    For Each oProperty in oMySchema.NamingProperties
        Response.Write "   " & oProperty & "<BR>"
    Next
End if

Response.Write "Members of the domain...<BR>"
For Each oMember in oMe
    Response.Write "   " & oMember.Name & "<BR>"
Next

%>

</BODY>
</HTML>
```

---

תחילה, הקוד מתאגד (Binds) ל-domain מוכר, במקרה זה ה-domain הוא Stellcom1 (אם ברצונך להפעיל תסריט זה, יהיה עליך לספק שם domain קביל אחר, כתחליף לשם זה). לאחר מכן, מאוחזרים ומוצגים המאפיינים המוגדרים באופן סטנדרטי ב-IADs. אחד ממאפיינים סטנדרטיים אלה הוא נתיב AdsPath לסכימה המשויכת ל-domain. הקוד משתמש בנתיב הסכימה כדי להתאגד (To Bind) לסכימה הממשית.

גם הסכימה עצמה היא מכולה (Container). הקוד מוודא עובדה זו, ולאחר מכן משתמש באוספים Containment, MandatoryProperties ו-OptionalProperties לבירור מספר פרטים אודות ממשקי ה-domain. תרשים 11.1 מציג את פלט דף ה-ASP.

ייתכן והבחנת כי קוד הדוגמה כולל את המשפט ON ERROR RESUME NEXT, וכמוהו גם מרבית הדוגמאות האחרות בפרק זה. משפט זה נחוץ, משום שמאפיינים רבים במודל האובייקטים של ADSI הם מאפייני רשות (Optional). ניסיון גישה למאפייני רשות שאינם כלולים ביישום (Implementation) זה או אחר של ADSI יוביל לשגיאות שיש להתעלם מהן.




**תרשים 11.1** הצגת סכימה

מתוך עיון בפלט זה, ניתן לראות כי תמיכת ADSI ב- WinNT Domains כוללת את מאפייני הרשות הבאים, המוצגים בטבלה 11.6.

**טבלה 11.6** מאפיינים המשויכים ל-Domain

| מאפיין | תיאור |
|---|---|
| MinPasswordLength | על הסיסמה להכיל לפחות מספר זה של תווים |
| MinPasswordAge | מספר השניות שעל סיסמה להתקיים, לפני שניתן לשנות אותה |
| MaxPasswordAge | מספר השניות שמותר לסיסמה להתקיים, לפני שיש לשנות אותה |

| מאפיין | תיאור |
|---|---|
| MaxBadPasswordsAllowed | מספר ניסיונות התחברות כושלים למערכת, בפרק זמן המוגדר במאפיין LockoutObservationInterval, לפני שהחשבון ייעל |
| PasswordHistoryLength | מספר הסיסמאות הייחודיות הנדרש לפני שניתן להשתמש בסיסמה כלשהי פעם נוספת |
| AutoUnlockInterval | מספר השניות לנעילת החשבון, במידה ומספר ניסיונות ההתחברות הכושלים עולה על זה המוגדר במאפיין MaxBadPasswordsAllowed |
| LockoutObservationInterval | טווח הזמן בשניות בו צריכים להתבצע מספר ניסיונות התחברות כושלים (כפי שמוגדר במאפיין MaxBadPasswordsAlowed) לפני שהחשבון ינעל |

על ידי שימוש בחלק ממאפייני WinNT namespace domain, ניתן לכתוב קוד ASP פשוט לשינוי מדיניות התיישנות הסיסמאות (Password Aging Policy).

הקוד בתדפיס 11.4 מדגים אחת מהדרכים האפשריות. ניתן למצוא את הקוד גם בתקליטור המצורף, בקובץ AgePasswords.asp.

**הערה:**

כדי שתסריט זה יפעל כראוי, על דף ASP שבתדפיס 11.4 להיות מוגן בסיסמה. כדי ללמוד כיצד להגן בסיסמה על דף ASP, ראה פרק 23. כמו כן, יהיה עליך להחליף את ה- Domain בשם ה-domain של ה-Windows NT שלך.

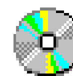
**תדפיס 11.4** מדיניות התיישנות סיסמאות

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" CONTENT="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<%

Dim oDomain

'On Error Resume Next

Response.Write "Invoking strict password security...<BR>"
```

```
'bind to a domain object
'need to be a domain admin to execute this script
Set oDomain = GetObject( "WinNT://Domain" )

'cache up the changes
oDomain.MinPasswordLength = 8
oDomain.MinPasswordAge = 60 * 60 * 24 * 2
oDomain.MaxPasswordAge = 60 * 60 * 24 * 30

'flush the changes
oDomain.SetInfo

Response.Write "All done...<BR>"
%>

</BODY>
</HTML>
```

# טיפול במשתמשים וקבוצות

הדוגמה הקודמת הביאה למיסוד מדיניות סיסמאות גלובלית לכל domain. מטלה נפוצה נוספת שעל מנהל מערכת (System Administrator) לבצע היא יצירה וניהול של קבוצות. ADSI מאפשרת יצירת יישומים למיכון מטלות הכרחיות (אם כי שכיחות למדי) אלו.



הדוגמה בתדפיס 11.5 מפרטת את משתמשי ה-domain על ידי איגוד (Binding) לקבוצת המשתמשים המשויכת ל-domain של WinNT. ניתן למצוא את הקוד גם בתקליטור המצורף, בקובץ ListUsers.asp.

**הערה:**

כדי שתסריט זה יפעל כראוי, על דף ASP שבתדפיס 11.5 להיות מוגן בסיסמה. כדי ללמוד כיצד להגן בסיסמה על דף ASP, ראה פרק 23. כמו כן, יהיה עליך להחליף את Server בשם שרת ה-Windows NT שלך.

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" CONTENT="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<%

Dim oUsersGroup
Dim oUser
Dim oMember

Set oUsersGroup = GetObject("WinNT://server/Users")

Set oMembers = oUsersGroup.Members()
For Each oUser in oMembers
    Response.Write oUser.Name & "<BR>"
Next

%>

</BODY>
</HTML>
```

דוגמה זו מציגה את שמות המשתמשים כטקסט HTML. השימוש ב-ASP מאפשר גם יצירה פשוטה של רשימה נפתחת של המשתמשים, על ידי שימוש במשפט SELECT של HTML.

ניתן להשתמש ב-ADSI גם למטרת ניהול חשבונות משתמש. כדי ליצור משתמש חדש, יש לאגד (To Bind) למחשב או PDC, ולאחר מכן להשתמש בשיטה Create המסופקת על ידי האובייקט IADsContainer.

הדוגמה בתדפיס 11.6 יוצרת משתמש, ומאתחלת את סיסמתו הראשונית ל-12345678. ניתן למצוא את הקוד גם בתקליטור המצורף, בקובץ CreateUser.asp.

**הערה:**

כדי שתסריט זה יפעל כראוי, על דף ASP שבתדפיס 11.6 להיות מוגן בסיסמה. כדי ללמוד כיצד להגן בסיסמה על דף ASP, ראה פרק 23. כמו כן, יהיה עליך להחליף את Server בשם שרת ה-Windows NT שלך.

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" CONTENT="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<%

Dim oDomain
Dim oUser

Response.Write "Creating a new user...<BR>"

'bind to a computer or PDC
Set oDomain = GetObject("WinNT://server")

'create the user need to be a domain admin to execute this script
Set oUser = oDomain.Create("User", "newADSIuser")

'persist the user
oUser.SetInfo

'assign a default password
oUser.SetPassword("12345678")

'persist the password
oUser.SetInfo

Response.Write "Done...<BR>"

%>

</BODY>
</HTML>
```

קטע הקוד שבתדפיס עושה שימוש בממשק ADSI נוסף - האובייקט User. אובייקט זה הוא ניסיון להפוך לתקן (standard) את סוגי המידע השונים המתוחזקים על ידי Directory Service, אודות כל משתמש. הסכימה ניתנת להרחבה, כמובן. ההפשטה User של ADSI לא דורשת שימוש באף אחד מהמאפיינים הזמינים בה. כולם מאפייני רשות (Optional). טבלה 11.7 מפרטת את השיטות והמאפיינים הזמינים דרך ממשק User של ADSI.

**טבלה 11.7** שיטות ומאפיינים ב-IADsUser

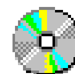
* מקרא: **קב** - **ק**ריאה **ב**לבד, **קכ** - **ק**ריאה **וכ**תיבה

| שיטה/מאפיין | תיאור |
|---|---|
| BadLoginAddress | כתובת הצומת האחרונה הנחשדת כפולשת (קב) |
| BadLoginCount | מספר ניסיונות ההתחברות הכושלים מאז האתחול האחרון (קב) |
| LastLogin | תאריך ושעת ההתחברות האחרונה (קב) |
| LastLogoff | תאריך ושעת [ההתנתקות] האחרונה (קב) |
| LastFailedLogin | תאריך ושעת ניסיון ההתחברות הכושל האחרון (קב) |
| PasswordLastChange | תאריך ושעת שינוי הסיסמה האחרון (קב) |
| Description | טקסט המתאר את המשתמש (קכ) |
| Division | החטיבה בתוך הארגון (קכ) |
| Department | יחידה ארגונית בתוך הארגון (קכ) |
| EmployeeID | זיהוי העובד של המשתמש (קכ) |
| FullName | שמו המלא של המשתמש (קכ) |
| FirstName | שמו הפרטי של המשתמש (קכ) |
| LastName | שם המשפחה של המשתמש (קכ) |
| OtherName | שם נוסף (כינוי, שם אמצעי) של המשתמש (קכ) |
| NamePrefix | תואר שם המשתמש (.Mr., .Mrs, .Dr וכדומה) (קכ) |
| NameSuffix | סיומת שם המשתמש (.Jr, הבן, II וכדומה) (קכ) |
| Title | תואר המשתמש (קכ) |
| Manager | מנהלו של המשתמש (קכ) |
| TelephoneHome | מספר הטלפון הביתי של המשתמש (קכ) |
| TelephoneMobile | מספר הטלפון הנייד של המשתמש (קכ) |
| TelephoneNumber | מספר הטלפון של המשתמש בעבודה (קכ) |
| TelephonePager | מספר האיתורית של המשתמש (קכ) |
| FaxNumber | מספר הפקס של המשתמש (קכ) |
| OfficeLocations | מערך של כתובות משרד (קכ) |
| PostalAddresses | מערך של כתובות דואר (קכ) |

| שיטה/מאפיין | תיאור |
|---|---|
| PostalCodes | מערך של מספרי מיקוד (קכ) |
| SeeAlso | מערך של נתיבי AdsPath לאובייקטים משויכים (קכ) |
| AccountDisabled | מצב החשבון של המשתמש (קכ) |
| AccountExpirationDate | תאריך ושעה בהם יחסמו ניסיונות התחברות נוספים של המשתמש (קכ) |
| GraceLoginsAllowed | מספר הפעמים בהן מותר למשתמש להתחבר לאחר פקיעת תוקף הסיסמה (קכ) |
| IsAccountLocked | מצב נעילת החשבון של משתמש (קכ) |
| LoginHours | הזמנים ביום ובשבוע שהמשתמש יכול להתחבר (קכ) |
| LoginWorkstations | רשימה של תחנות עבודה מהן מותר למשתמש להתחבר (קכ) |
| MaxLogins | מספר החיבורים המותרים למשתמש בו זמנית (קכ) |
| MaxStorage | נפח הדיסק המקסימלי המוקצה למשתמש (קכ) |
| PasswordExpirationDate | תאריך ושעת פקיעת תוקף הסיסמה של המשתמש (קכ) |
| PasswordMinimumLength | מספר התווים המינימלי שעל סיסמת המשתמש להכיל (קכ) |
| PasswordRequired | האם דרושה סיסמה עבור משתמש זה (קכ) |
| RequireUniquePassword | האם על הסיסמה החדשה להיות שונה מסיסמאות קודמות ברשימת הסיסמאות של המשתמש (קכ) |
| EmailAddress | כתובת הדואר האלקטרוני של המשתמש (קכ) |
| HomeDirectory | תיקיית הבית של המשתמש (קכ) |
| Languages | מערך של שפות עבור המשתמש (קכ) |
| Profile | נתיב לפרופיל המשתמש (קכ) |
| LoginScript | נתיב לתסריט ההתחברות של המשתמש (קכ) |
| Picture | מערך של בתים המכיל תמונה של המשתמש (קכ) |
| HomePage | כתובת URL לאתר הבית של המשתמש (קכ) |
| Groups() | מחזיר את הקבוצות להן שייך המשתמש |
| SetPassword() | קובע את סיסמת המשתמש |

| תיאור | שיטה/מאפיין |
|---|---|
| משנה את סיסמת המשתמש מהערך הישן המצוין לערך החדש | ChangePassword() |

כפי שניתן לראות, יישום סטנדרטי של ADSI כולל אוסף מקיף מאד של מאפיינים לתיאור משתמש.



מטלה נוספת שעל מנהל מערכת (System Administrator) לבצע לעיתים תכופות היא הוספה והסרה של משתמשים מקבוצות. ADSI מקלה על המתכנתים את עבודת הפיתוח של כלי ניהול (Administration Tools) מבוססי אינטרנט (Web Base) לסיוע בביצוע מטלות אלו. תדפיס 11.7 מדגים שימוש ב-ADSI לניהול קבוצות. ניתן למצוא את הקוד גם בתקליטור המצורף, בקובץ GroupLogic.asp.

**תדפיס 11.7** ניהול קבוצות

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" CONTENT="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<%

Dim oDomain
Dim oGroup
Dim oMember
Dim oMembers
Dim bMember
Dim strGroup

On Error resume next
Set oDomain = GetObject("WinNT://server")

Set oGroup = GetObject("WinNT://server/Engineers")

If err <> 0 Then
     ' create a new group called Engineers
     Set oGroup = oDomain.Create("group", "Engineers")
     oGroup.SetInfo
End If

' add members to the group
oGroup.Add("WinNT://server/user1")

' remove members from the group
bMember = oGroup.IsMember("WinNT://server/Users/user1")
If bMember Then
    oGroup.Remove("WinNT://server/Users/user1")
End If
```

```
' list all members of the group
Response.Write "Members of the Engineering group...<BR>"
set oMembers = oGroup.Members()
For Each oMember in oMembers
    Response.Write "   " & oMember.Name & "<BR>"
Next

' delete the group - assumes a group call lackies...
' construct a possible ADsPath based on computer/domain and group name
strGroup = oDomain.AdsPath & "/" & "Lackies"

'attempt to bind to the group
Set oGroup = GetObject(strGroup)
if Err Then
    Response.Write "Group not found...<BR>"
Else
    ' Get parent of group
    Set oParent = GetObject(oGroup.Parent)
    if Err Then
        oParent.Delete "group", oGroup.Name
    End if
End if

%>

</BODY>
</HTML>
```

דוגמה זו בודקת האם קיימת קבוצה בשם Engineers. אם הקבוצה לא קיימת, היא נוצרת על ידי התסריט. כשהקבוצה קיימת, מתווסף לה משתמש בשם User1. התסריט מוודא שהמשתמש אכן התווסף לקבוצה, ומייד מסיר אותו ממנה (זהו קוד למטרת הדגמה. אין כל שימוש להוספת משתמש והסרתו מייד לאחר מכן, מלבד להדגמת האפשרות).

לאחר מכן, מראה התסריט כיצד לעבור על חברי הקבוצה בזה אחר זה ולהציג את שמותיהם. לבסוף, מוצגת האפשרות למחיקת קבוצה על ידי שימוש בשיטה Delete של ההורה שלה.

זכור, עליך לספק שם שרת תקין ולהפעיל את התסריט תחת ההרשאות המתאימות, כדי שתוכל לתפעל קבוצות באופן מוצלח.

# שליטה בשירותי NT

ניתן להשתמש ב-ADSI כדי להתחיל ולעצור שירותים של Windows NT. הדבר מאפשר יצירה של ממשקי רשת המסוגלים לבצע כל דבר, החל משליטה בפעולתה של חוות שרתים, וכלה בעצירת שירות FTP בתחנת העבודה הביתית שלך.

**אזהרה:**

השליטה בשירותי NT עלולה להיות מסוכנת. ניתן בנקל לכתוב דף ASP העושה שימוש ב-ADSI וגורם לעצירתם של כל השירותים. אפילו יותר גרוע - ניתן לכתוב תסריט שישנה את מאפייני ברירת מחדל שבאתחול (Default Startup Properties) של אחד או יותר מהשירותים, דבר שיגרום לכך שהמחשב לא יתפקד כראוי החל מהפעם הבאה שתאתחל אותו. הזהר. אל תנסה לשלוט בשירותים אלא אם אתה יודע מה אתה עושה.

ADSI מספק את האפשרות לביצוע כמעט כל משימה הזמינה באמצעות היישומון Services בלוח הבקרה. המפתח לשליטה בשירותי NT נמצא בממשק IADsService. טבלה 11.8 מזהה את המאפיינים והשיטות של ממשק זה.

**טבלה 11.8** שיטות ומאפיינים בIADsService

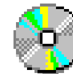
| שיטה/מאפיין | תיאור |
|---|---|
| Dependencies | רשימה של שירותים אחרים שצריכים לפעול כדי שיהיה ניתן להתחיל שירות זה |
| DisplayName | שם השירות המוצג למשתמש |
| ErrorControl | רמת האבטחה שתופעל במידה והשירות לא מצליח להתחיל |
| HostComputer | נתיב AdsPath למחשב המארח |
| LoadOrderGroup | סדר הטעינה לשירות זה |
| Path | נתיב לקובץ ההפעלה של השירות |
| ServiceAccountName | שם החשבון המשמש להפעלת השירות |
| ServiceAccountPath | נתיב ה-AdsPath של החשבון המשמש להרצת השירות |
| ServiceType | אחד מארבעה סוגי שירותים |
| StartType | אחד מחמישה ערכים המתארים כיצד יותחל השירות |
| StartupParameters | פרמטרים המועברים לשירות בעת האתחול |
| Version | גרסת השירות |

הממשק IADsService חושף את כל המאפיינים הנקבעים באופן רגיל על ידי היישומון Services בלוח הבקרה. ממשק ADSI אחר מספק את השליטה במצבם התפעולי (Operational State) של השירותים. כדי לעצור, להפעיל, להשהות ולהפעיל מחדש שירותים, יש להשתמש בממשק IADsServiceOperations. טבלה 11.9 מפרטת את השיטות הנחשפות בממשק IADsServiceOperations.

**טבלה 11.9** שיטות ומאפיינים ב-IADsServiceOperations

| שיטה/מאפיין | תיאור |
|---|---|
| Status | מחזיר את מצבו הנוכחי של השירות |
| Start | הפעלת השירות |
| Stop | עצירת השירות |
| Pause | השהיית השירות |
| Continue | המשך השירות |
| SetPassword | קביעת הסיסמה המשויכת עם השירות |

השליטה בשירותים דרך ADSI היא פשוטה למדי. הדוגמה בתדפיס 11.8 מדגימה כיצד לעצור ולהפעיל שירות. ביסודו, בודק התסריט את מצב השירות W3SVC, ומשנה את מצב השירות למצב המנוגד. אם השירות פעיל (Running) התסריט עוצר אותו. אם השירות במצב של עצירה (או השהיה), גורם לו התסריט להתחיל מחדש (Restart).


ניתן למצוא את קוד התדפיס גם בתקליטור המצורף, בקובץ Services.asp.

## הערה:

כדי שתסריט זה יפעל כראוי, על דף ASP שבתדפיס 11.8 להיות מוגן בסיסמה. כדי ללמוד כיצד להגן בסיסמה על דף ASP, ראה פרק 23. כמו כן, יהיה עליך להחליף את Server בשם שרת ה-Windows NT שלך.

**תדפיס 11.8** שליטה בשירות של Windows NT.

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" CONTENT="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<%

Dim oService

'Warning this will shut off your www service, causing early termination of script
Set oService = GetObject("WinNT://server/W3SVC")
Response.Write oService.name &",  "& oService.class &"<BR>"
```

```
Select Case oService.Status
    Case 1:  'stopped
        oService.Start
    Case 4:  'running
        oService.Stop
    Case 7:  'paused
        oService.Continue
End Select

%>

</BODY>
</HTML>
```

# שליטה במדפסות

ADSI מספק גם שליטה תכנותית על ניהול מדפסות ותורי הדפסה. על ידי שימוש ב-ADSI ניתן לגלות אילו מדפסות זמינות, ומה מצב תור ההדפסה שלהן. ניתן להפעיל, לעצור ולהשהות את תור ההדפסה. ADSI מאפשר גם תפעול של עבודות הדפסה (Print Jobs) יחידות. טבלה 11.10 מציגה חלק מהמאפיינים והשיטות שבממשק IADsPrintQueue.

**טבלה 11.10** שיטות ומאפיינים ב-IADsPrintQueue

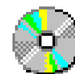
| מאפיין | תיאור |
|---|---|
| PrintPath | נתיב AdsPath למדפסת המשותפת |
| Model | שם התקן המדפסת המשויך עם התור |
| Description | תאור מילולי של תור ההדפסה |
| Location | מחרוזת טקסט המציינת את מיקום תור ההדפסה |
| DefaultJobPriority | עדיפות ברירת המחדל עבור כל עבודת הדפסה |
| BannerPage | נתיב מערכת קבצים לעמוד מפריד |
| PrintDevices | מערך מחרוזות BSTR המגדירות את התקני ההדפסה המשויכים לתור |
| NetAddress | מערך שדות כתובות רשת |
| Status() | מקבל את מצב (Status) תור המדפסת |
| PrintJobs() | מחזיר את רשימת ה-IADSPrintJobs שבתור |
| Pause() | משהה את עיבוד תור ההדפסה |
| Resume() | ממשיך את עיבוד תור ההדפסה |
| Purge() | מנקה את תור ההדפסה |

טבלה 11.11 מציגה חלק מהמאפיינים והשיטות שבממשק IADsPrintJob.

**טבלה 11.11** שיטות ומאפיינים ב-IADsPrintJob

| מאפיין | תיאור |
|---|---|
| HostPrintQueue | נתיב AdsPath המשויך לתור ההדפסה |
| User | שם המשתמש של האדם שהגיש את עבודת ההדפסה |
| UserPath | נתיב AdsPath לאדם שהגיש את עבודת ההדפסה |
| TimeSubmitted | הזמן בו הוגשה עבודת ההדפסה |
| TotalPages | מספר העמודים בעבודת ההדפסה |
| Size | גודל עבודת ההדפסה בבתים |
| Description | טקסט המתאר את עבודת ההדפסה |
| Priority | עדיפות עבודת ההדפסה |
| StartTime | הזמן המוקדם ביותר בו יכולה עבודת ההדפסה להתחיל |
| UntilTime | הזמן המאוחר ביותר בו יכולה עבודת ההדפסה להתחיל |
| Notify | שם האדם שיש להודיע לו לאחר השלמת עבודת ההדפסה |
| NotifyPath | נתיב AdsPath של האדם שיש להודיע לו לאחר השלמת עבודת ההדפסה |
| Status() | מצב עבודת ההדפסה |
| TimeElapsed() | מספר השניות מאז תחילת עבודת ההדפסה |
| PagesPrinted() | מספר העמודים שהודפסו |
| Position() | מיקום מספרי של עבודת ההדפסה בתור |
| Pause() | משהה את עיבוד עבודת ההדפסה |
| Resume() | ממשיך עיבוד עבודת ההדפסה |

כפי שניתן לראות, מספק ADSI שליטה רבה בעבודות ומשאבי ההדפסה שונים הזמינים בסביבת הרשת.



הקוד בתדפיס 11.9 מדגים שימוש ב-ADSI לשליטה במשאבי ההדפסה. ניתן למצוא את הקוד גם בתקליטור המצורף, בקובץ Printers.asp.

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" CONTENT="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<%

Dim oServer
Dim oPrintQueue
Dim strFilter

strFilter = Array("PrintQueue")
Set oServer = GetObject("WinNT://server")
oServer.Filter = strFilter

Response.Write "Printers on " & oServer.Name & "<BR>"
For Each oPrintQueue in oServer
    Response.Write "   " & oPrintQueue.Name & "<BR>"
    oPrintQueue.Purge
Next

%>

</BODY>
</HTML>
```

הדוגמה עוברת על תורי ההדפסה הקיימים בשרת המצוין בזה אחר זה (זכור לציין את שם השרת שלך לפני הפעלת התסריט). שמה של כל מדפסת מוצג לדפדפן הלקוח. לאחר מכן מנוקה (Purged) תור ההדפסה מכל העבודות.

# ניהול שרת Exchange

שרת Exchange בגירסה 5.5 משתמש ב-LDAP למעקב אחר נמענים (חשבונות משתמש). ניתן להשתמש ב-ADSI כדי לגשת למחסן ה-LDAP של Exchange באמצעות תוכנה. דבר זה מאפשר למתכנתים קריאה וכתיבה של פרמטרים רבים הנוגעים לנמען, ביניהם נתונים כגון שמו, כתובתו, מספרי הטלפון בעבודה ובבית, זימונית, פקס ומספר הטלפון הסלולרי של משתמש Exchange.

ממשק LDAP מספק גם גישה להירכיות של תיקיות ציבוריות, קבוצות, ורשימות תפוצה, כמו גם הגדרות של POP3, HTTP, IMAP ו-NNTP. כמעט כל פעולה הניתנת לביצוע באמצעות ממשק המשתמש של Exchange, ניתן לבצע גם באמצעות סכימת LDAP, והממשקים והאובייקטים המשויכים אליה.

בהמשך פרק זה, נביא דוגמה פשוטה לגישה לנתוני נמענים בשרת Exchange דרך ADSI.

# ADSI ו-Site Server

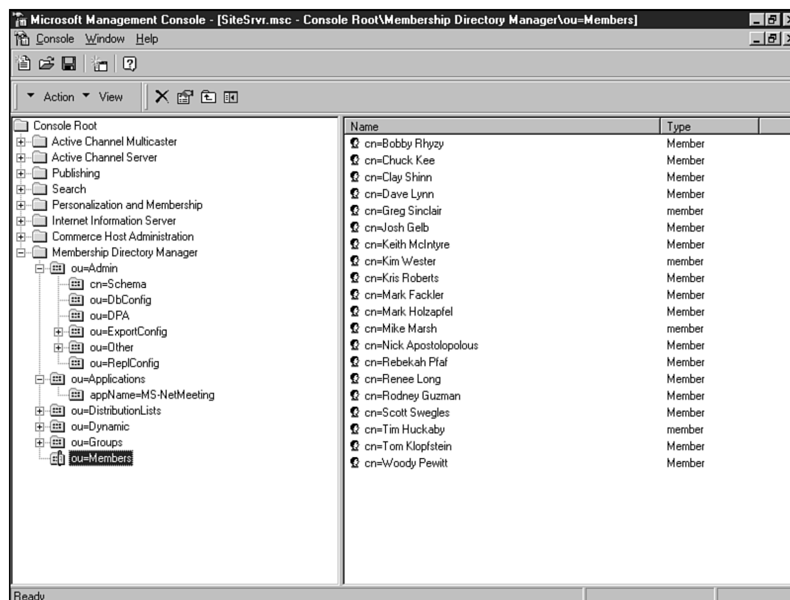
דוגמה נוספת לאופן האימוץ של ADSI ו-LDAP על ידי Microsoft, ניתן למצוא בגירסה 3.0 של Site Server. Site Server היא פלטפורמה חזקה מאד לפיתוח מערכות אינטרנט הכוללות רשתות פנימיות (אינטראנט, Intranets), רשתות חיצוניות (אקסטראנט, Extranets) ואתרי סחר אלקטרוני. רכיבי החברות (Membership) וההתאמה האישית (Personalization) של Site Server מבוססים על LDAP (למעשה, Site Server Membership מאפשר שימוש הן באימות חברות של Windows NT, והן בזה של LDAP). אך תהה שיטת האימות אשר תהה, ניתן להשתמש ב-LDAP לתחזוקת היבטים רבים של Site Server, ביניהם מילונים של אתר (Site Vocabularies), מידע סכימה אודות קבוצות וחברים, פרמטרים והגדרות ליישומים ספציפיים, ורשימות תפוצה של Direct Mail.

היתרון בשימוש ב-LDAP הוא ההתאמה הנוחה למיליוני משתמשים. בשימוש ביישומי אינטרנט, מאפשר LDAP כתיבה של יישומי רשת מאובטחים, שאינם מוגבלים ליכולות האימות המוגבלות יחסית של Windows NT Domains.

יתרון נוסף של LDAP הוא יכולת השימוש ב-ADSI לביצוע שאילתות, ולשינוי תוכן ה-Directory על ידי תכנות. בנוסף, מספק ממשק אובייקט המשתמש האקטיבי (Active User Object, AUO) ממשק תכנותי נוסף לעץ ה-Directory. AUO משתמש ב-ADSI, ולמעשה חושף את הממשקים IADs ו-IADsContainer, במנגנון גישה מופשט ל-Directory Tree.

Site Server מיישם Directory Tree מיוחד, המאפשר שליטה בהיבטים רבים של האתר. העץ מתחיל עם שם הארגון כשורש. מתחת לארגון, נמצאות מספר יחידות ארגוניות (OU) : Admins, Groups, Members, Applications ו-DistributionLists. היחידה הארגונית Admins כוללת מספר יחידות ארגוניות כפופות (Subordinate OUs), ביניהן היחידה הארגונית Schema.

ניתן להציג, לשנות, לעיין ולהרחיב את העץ הארגוני של LDAP, על ידי שימוש בקונסולת הניהול של Microsoft (MMC). תרשים 11.2 מציג הרחבה חלקית של הצומת Personalization and Membership במחסן LDAP של Site Server.




**תרשים 11.2** גישה ל-LDAP Directory של השרת על ידי שימוש ב-MMC

היחידה הארגונית Members כוללת יחידה ארגונית כפופה בשם AnonymousUsers. מכולה (Container) זו משמשת לאגירת נתונים אודות משתמשים שאינם חברים באתר, אך מאפשרים אימות באמצעות Cookies.

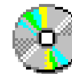
ניתן לאסוף ולתחזק מידע התאמה אישית אודות משתמשים כאלה במכולה O=XXX, OU=Members, OU=AnonymousUsers עד להרשמתם לאתר. לאחר ההרשמה, ניתן יהיה להעתיק את נתוני ההתאמה האישית לאובייקטי עלה (Leaf Objects) המייצגים את החברים באתר.

המכולה OU=Members משמשת לאחסון נתוני כל החברים. כל חבר מיוצג על ידי צומת עלה (Leaf Node) בתוך מכולת החברים. השדה cn (שם כללי - Common Name) באובייקט Member משמש להבדלה בין משתמשים. מכאן נובע כי השם הנבדל (Distinctive Name) עבור חבר הוא cn=Username, OU=Members, O=Organization.

דוגמה לתחביר כללי זה יכולה להיות:

cn=kmcintyre, OU=members, o=stellcom

תצורת LDAP של Site Server תומכת בממשקי ADSI הבאים: IADs, IADsContainer, IADsClass, IADsProperty ו-IADsOpenDSObject. נתקלת כבר בכל אחת ממחלקות אלו, מלבד IADsOpenDSObject, בה נדון מאוחר יותר בפרק זה. ממשקים בסיסיים אלה מאפשרים ליישום לגלות את המאפיינים הייחודיים המשויכים למשתמש (באמצעות IADsClass והסכימה), למנייה ומציאת משתמשים (באמצעות הממשק IADsContainer), לפתוח אובייקט משתמש (באמצעות הממשק IADsOpenDSObject) ולגשת למאפיינים המגדירים את פרופיל המשתמש (באמצעות IADsProperty).



הקוד בתדפיס 11.10 מדגים גישה לאובייקט של חבר ב-Directory Tree, ולאחר מכן מעבר על כל המאפיינים המקושרים עם המשתמש בזה אחר זה. ניתן למצוא את הקוד גם בתקליטור המצורף, בקובץ SiteServer.asp.

**תדפיס 11.10** גישה למאפייני חברות של SS3 דרך ADSI

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" CONTENT="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<%

Dim oNamespace
Dim oMember

'bind to the namespace
Set oNamespace = GetObject("LDAP:")

'open the member object using authentication credentials credentials must
'be meaningful in terms of the service being provided i.e. fully qualified
'path to the admin user
Set oMember = oNamespace.OpenDSObject(
    ↳"LDAP://server/cn=user,
    ↳ou=Members, o=organization", "cn=adminUser,ou=Members, o=organization",
    ↳"password", 0)

'now access some of the members properties
curBalance = oMember.AccountBalance
If curBalance > 1000 Then
    oMember.bHighRoller = 1
Else
    oMember.bHighRoller = 0
End if

%>

</BODY>
</HTML>
```

# שימוש ב-ADO עם ADSI

קיימות דרכים רבות לביצוע חיפוש ב-Directory Services. טכניקת החיפוש הפשוטה ביותר תהיה למנות (Enumerate) את כל האובייקטים שב-IADsContainer, ולאחר מכן לבחון כל אחד מהם מול הקריטריון הרצוי. גישה ברוטלית שכזו עלולה להוביל לביצועים גרועים ביותר, אך היא קלה ליישום ויכולה להיות שימושית, אם ידוע כי מספר הרשומות הקיימות קטן.

חלופה לטכניקה ברברית זו, יכולה להיות שימוש בפילטר ADSI בעת המנייה (Enumeration) של IADsContainer. פילטר זה הוא מערך של BSTR (סוג של טיפוס נתונים המייצג מחרוזת של תווים), המציין את סוגי האובייקטים שברצונך שיוחזרו.

תדפיס 11.11 מדגים את אופן השימוש במאפיין IADsContainer::Filter להקטנת תחום המנייה. ניתן למצוא את הקוד גם בתקליטור המצורף, בקובץ Filter.asp.

**תדפיס 11.11** שימוש במאפיין Filter של IADsContainer

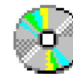
```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" CONTENT="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<%

Dim oContainer
Dim oRecipient

'bind to the exchange server
Set oContainer = GetObject("LDAP://server/cn=Recipients,ou=site,o=organization")

'apply a filter
oContainer.Filter = Array("OrganizationalPerson")

'load the cache
oContainer.GetInfo

For Each oRecipient in oContainer
    'do something with the recipient\
    If oRecipient.postalCode = "92121" Then
        Response.Write oRecipient.Name & "<BR>"
    End if
Next

%>

</BODY>
</HTML>
```

דוגמה זו שראינו זה עתה, ניגשת לשרת Exchange ועוברת בזה אחר זה על כל הנמענים שהם חברים בקבוצה OrganizationalPerson. המאפיין postalcode של כל נמען נבדק מול מיקוד (Zip Code) מסוים. אם מיקוד הנמען זהה ל-92121, מוצג שם הנמען בדפדפן הלקוח.

דרך נוספת לשיפור יעילות החיפושים היא שימוש במאפיין IADsContainer::hints, כדי להודיע ל-ADSI ולספק השירות אודות הפריטים אליהם אתה מעוניין לגשת. המאפיין hints מודיע לספק השירות לאלה מאפיינים ייגש היישום. כך יוכל ספק השירות לא לטעון למטמון (Cache) מאפיינים שאינם מעניינים את היישום. ספק השירות יוכל לטעון מאפיינים אלה ישירות מה-Persisted Store, אם וכאשר יהיה ליישום צורך בהם. במקרה בו יישום מתכוון לעבור על אובייקטים רבים בזה אחר זה, אך לבחון רק אחד מהמאפיינים של אובייקטים אלה, מהירות החיפוש תגדל אם תשתמש ב-hints ותודיע לספק השירות מהו המאפיין שייבדק על ידי היישום (לא כל ספקי השירות מכבדים את המאפיין hints).

הדרך היעילה ביותר להצגת שאילתות ל-Directory Services היא, כפי הנראה, על ידי שימוש ב-ADO. ADSI כולל ספק OLE DB, דרכו יכול ADO לבצע שאילתות אל מול Directory Service. זוהי שיטה רבת עוצמה למציאת אובייקטי עלה ספציפיים במרחב השמות. היא יכולה להיות מהירה בהרבה ממנייה של עץ הנתונים המלא של ה-Directory. ADO מספק גישת קריאה בלבד למחסן הנתונים. ספק OLE DB של ADSI אינו תומך בכתיבת ערכים לספריה דרך ADO. פרק 12 עוסק בשימוש ב-ADO וב-OLE-DB. תוכל להיעזר בו במקרה שלא תבין את קטע הקוד הבא.

השימוש ב-ADO לגישה ל-Directory TRee זהה לשימוש ב-ADO כנגד כל מחסן נתונים אחר. הקושי הגדול ביותר הוא הקישור הראשוני לספק ה-OLE DB של ADSI.



הקוד בתדפיס 11.12 מדגים את אופן הביצוע של שאילתה פשוטה באמצעות ADO. ניתן למצוא את הקוד גם בתקליטור המצורף, בקובץ ADOExample.asp.

**תדפיס 11.12** שימוש בADO להצגת שאילתה לADSI

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" CONTENT="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<%

Dim oADOConnection
Dim oRecordSet
Dim oMember
Dim bstrQuery
```

```
'establish the ADO connection
set oADOConnection = Server.CreateObject("ADODB.Connection")
oADOConnection.Provider = "ADSDSOObject"
oADOConnection.Open "ADs Provider"

'specify a query
bstrQuery =
"<LDAP://server/cn=Recipients,ou=site,o=organization>;(objectClass=*);adspath,cn;su
btree"

'execute the query
set oRecordset = oADOConnection.Execute( bstrQuery )
if err.number <> 0 Then Response.Write "error<BR>"

'access the records
Response.Write "Returned members <BR>"
Do While Not oRecordset.EOF
    Set oMember = GetObject( oRecordset.Fields(0).Value )
    Response.Write "   " & oMember.Name & "<BR>"
    oRecordset.movenext
Loop

%>

</BODY>
</HTML>
```

# שיקולי אבטחה

עד עתה, השתמשת ב-GetObject() כדי להתאגד (To Bind) לאובייקטי ADSI. גישה זו פועלת כל עוד ה-Directory Service אינו מוגן על ידי מנגנון אבטחה מבוסס ACL (Security ACLs). ADSI מספק מספר ממשקים לתמיכה וקידום של אבטחה. בסעיף זה, נעבור בזריזות על ממשקים אלה.

ADSI מספק את הממשק IADsOpenDSObject כאמצעי לאספקת אישורים (Credentials) בעת בקשת גישה ל-Directory Services. IADsOpenDSObject מכיל שיטה אחת שתהיה שימושית ללקוחות אוטומציה (Automation Clients) דוגמת יישומי ASP. שיטה זו היא OpenDSObject.

השיטה OpenDSObject() דורשת ארבעה פרמטרים :

```
OpenDSObject(DN, UserName, Password, Flags)
```

כאשר הפרמטר DN הוא נתיב ה-AdsPath של האובייקט שברצונך להתאגד (To Bind) אליו. UserName ו-Password מציינים את חשבון המשתמש באמצעותו יתבצע האגוד, ו-Flags יכול להיות ADSI_USE_SECURE_AUTHENTICATION או ADSI_USE_ENCRYPTION.

קריאה אופיינית ל-()OpenDSObject מוצגת בתדפיס 11.13.

**תדפיס 11.13** אימות באמצעות ()OpenDSObject

```
Set oNamespace = GetObject("WinNT:")
Set oDomain = oNamespace.OpenDSObject("WinNT://stellcom1",
   ➥"kmcintyre", "passthis", 0)

' use oDomain as we have before
Response.Write "Name = " & oDomain.Name & "<BR>"
```

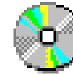
גרסת ADSI הנוכחית לא כוללת אובייקט לעבודה עם הרשאות קבצים שניתן להשתמש בו מתוך דפי ASP. אם ברצונך לתפעל הרשאות של קובץ על ידי שינוי רשימת בקרת הגישה (Access Control List - ACL) שלו, יהיה עליך להשתמש בתוכנת שורת הפקודה של DOS, Cacls.exe, או ברכיב מותאם במיוחד.

Cacls.exe הוא תוכנית שירות של DOS בה ניתן להשתמש לשינוי ACL של קובץ. התסריט בתדפיס 11.14 משתמש ב-Cacls.exe ובשיטה Run של האובייקט Shell של Windows Scripting Host לשינוי הרשאות קובץ בשם test.txt (דוגמה זו מתבססת על תסריט שהוצג במאמרם של Jeff Sanquist ו-Tom Moran, "Create a Virtual Directory Automatically with ADSI, ASP and a Couple of Magic Tricks, Part II". המאמר פורסם ב-Microsoft Web Workshop, בכתובת האינטרנט http://msdn.microsoft.com).

**תדפיס 11.14** קביעת הרשאות של קובץ באמצעות Cacls.exe

```
<%
currentUser = TRIM( Request.ServerVariables( "LOGON_USER" ) )
if currentUser = "" then
   Response.Status = "401 Not Authorized"
   Response.End
end if
acls = "cmd /c echo y| CACLS d:\test.txt"
acls = acls & " /g " & currentUser & ":f"
set objWSH = Server.CreateObject ( "WScript.Shell" )
result = objWSH.Run( acls, 0, True )
if result = 0 then
   Response.Write "Changed Permissions!"
else
   Response.Write "Problem"
end if
%>
```

תסריט זה כופה את הופעתה של תיבת דיאלוג של סיסמה, ולאחר מכן מעניק למשתמש שהתחבר הרשאות גישה מלאות לקובץ test.txt. השימוש בפרמטר g/ של Cacls.exe מעניק למשתמש את הזכויות לקובץ המצוין. למידע נוסף לגבי השימוש ב-Cacls.exe, ראה הערך Cacls.exe בעזרה של Windows NT.



השימוש בCacls.exe לשינוי הרשאות קבצים אינו אלגנטי או בר הרחבה במיוחד. פתרון טוב בהרבה יהיה השימוש ברכיב מותאם במיוחד. התקליטור המצורף לספר כולל גירסה מתפקדת לחלוטין של הרכיב SA-FileManager של חברת Software Artisan. הרכיב כולל שיטות לשינוי הרשאות של קבצים, ולהחזרת מידע לגבי בעלות על קבצים.

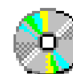
חלופה נוספת היא הרכיב AspAccessControl של חברת Persits Software, המאפשר, מלבד שינוי הרשאות קבצים, גם הצגה של הרשאות הקבצים הנוכחיות. למידע נוסף לגבי רכיב זה, פנה לאתר החברה: http://www.persits.com.

# שימוש ב-ADSI עם IIS MetaBase

Internet Information Server כולל אוסף של אובייקטים - אובייקטי IIS Admin - המיישמים ספק מרחב שמות ADSI ל-IIS. אובייקטים אלה מאפשרים לך לקבוע דרך תוכנה את תצורת מאפייני שרת האינטרנט שלך, כגון הרשאות הקבצים, התיקיות והיישומים שלו, תיקיות וירטואליות, ומסמכי ברירת מחדל. למעשה, כל פעולה הניתנת לביצוע ב-Internet Service Manager ניתן לבצע גם באמצעות אובייקטי IIS Admin (גרסת הHTML של Internet Service Manager, שMicrosoft כללה ב-IIS, עושה שימוש באובייקטי IIS Admin).

כשאתה משתמש באובייקטי IIS Admin לקביעת תצורת שרת האינטרנט שלך, אתה למעשה משנה ערכים במטא מסד של IIS (IIS Metabase). IIS משתמש במטא מסד, ולא ברישום המערכת, לאחסון נתוני התצורה שלו. נתיבי המפתחות במטא מסד מקבילים לנתיבי ADSI.

לדוגמה, התסריט בתדפיס 11.15 מחזיר את רשימת השמות והנתיבים של התיקיות הפיסיות והווירטואליות הממוקמות תחת אתר ברירת המחדל (Default Web Site). התסריט עושה שימוש בנתיב ה-ADSI IIS://Server/W3SVC/1/Root, לאחזור שורש התיקיה הווירטואלית של אתר ברירת המחדל שלך (אתר מספר 1).



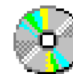
ניתן למצוא את הקוד גם בתקליטור המצורף, בקובץ IISListVDirs.asp.

**תדפיס 11.15** פירוט התיקיות באתר ברירת המחדל

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" CONTENT="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<%

ON ERROR RESUME NEXT

myServer = Request.ServerVariables( "SERVER_NAME" )
Response.Write "Server = " & myServer
ADSIPath = "IIS://" & myServer & "/W3SVC/1/ROOT"
Set defaultSite = getObject( ADSIPath )
for each thing in defaultSite
  Response.write "<P><B>" & thing.Name & "</B>"
  Response.write "<BR>" & thing.Class
  Response.write "<BR>" & thing.Path
next

%>

</BODY>
</HTML>
```

התסריט בתדפיס 11.15 יוצר באופן אוטומטי נתיב ADSI לתיקיית השורש הווירטואלית של אתר ברירת המחדל, על ידי אחזור הפריט SERVER_NAME מתוך האוסף ServerVariables. לאחר מכן, נעשה שימוש בלולאת FOR...EACH, להצגת כל התיקיות הפיסיות והווירטואליות הממוקמות תחת התיקיה הווירטואלית של אתר ברירת המחדל.

ניתן להשתמש באובייקטי IIS Admin גם ליצירת תיקיה וירטואלית חדשה וקביעת מאפייניה. התסריט בתדפיס 11.16 יוצר תיקיה פיסית בשם testdir, ומשייך אותה לתיקיה וירטואלית חדשה בשם StellcomScripts.


ניתן למצוא את הקוד גם בתקליטור המצורף, בקובץ IISCreateVDir.asp.

**תדפיס 11.16** יצירת תיקיה וירטואלית חדשה

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" CONTENT="Microsoft Visual Studio 6.0">
</HEAD>
```

```
<BODY>

<%

' Create the Physical Directory
Set FS = Server.CreateObject( "Scripting.FileSystemObject" )
FS.CreateFolder "c:\testdir"

' Create the Virtual Directory
myServer = Request.ServerVariables( "SERVER_NAME" )
ADSIPath = "IIS://" & myServer & "/W3SVC/1/ROOT"
Response.Write "ADSIPath = " & ADSIPath
Set defaultSite = getObject( ADSIPath )
Set vDir = defaultSite.Create( "IISWebVirtualDir", "StellcomScripts" )
vDir.Path = "c:\testdir"

' Set Virtual Directory Properties
vDir.AccessRead = TRUE
vDir.AccessScript = TRUE
vDir.DefaultDoc = "index.asp"
vDir.SetInfo
Response.Write "Virtual Directory Created!"

%>

</BODY>
</HTML>
```

התסריט בתדפיס 11.16 משתמש ברכיב File Access (רכיב זה נידון בפרק 8) ליצירת תיקיה פיסית חדשה בשם testdir. לאחר מכן, נעשה שימוש בנתיב IIS://Server/W3SVC/1/Root של ADSI לאחזור אובייקט המייצג את תיקיית השורש הווירטואלית. השיטה Create של האובייקט IADsContainer משמשת ליצירת תיקיה וירטואלית חדשה בשם StellcomScripts, ולאחר מכן נקבעים עבור תיקיה זו הרשאות כתיבה, הרשאות תסריט, ומסמך ברירת מחדל. לבסוף, נשמרים מאפייני התיקיה הווירטואלית החדשה במטא מסד של IIS (IIS Metabase), באמצעות קריאה לשיטה SetInfo של ADSI.

# סיכום

בפרק זה, הוצגו בפניך חלק מהסיבות שהובילו ליצירת ADSI. למדת כיצד להשתמש ב-ADSI לביצוע מטלות המבוצעות, בדרך כלל, על ידי מנהל המערכת. בחנת רבים ממשקי האוטומציה הנפוצים של ADSI - אלה הניתנים לשימוש בסביבת ASP. ראית דוגמאות לקוד לניהול משתמשים, קבוצות, שירותים, מדפסות ועבודות הדפסה. כמו כן, בחנת בקצרה את חשיבות ADSI ו-LDAP לשליטת התוכנית בשרת Exchange וב-Site Server. הפרק כיסה גם את ממשקי האבטחה הזמינים דרך ADSI, ואת אופן השימוש ב-ADSI לקביעת תצורת IIS מתוך תסריטי דפי ASP.

# חלק 3

# שימוש בדפי ASP עם מסדי נתונים

**בחלק זה:**

- עבודה עם Connections ו-Data Sources
- עבודה עם Recordsets
- עבודה עם האובייקט Command
- עבודה עם Index Server וחיפוש טקסט-מלא

## פרק 12

# עבודה עם Connections ונתונים

**בפרק זה:**

- **מילון מושגי גישה לנתונים**
- **יצירת Connections עם OLE DB ו-ODBC**
- **חיבור לשרת SQL של Microsoft**
- **חיבור למסד הנתונים Microsoft Access**
- **הפעלת משפטי SQL באמצעות האובייקט Connection**
- **שיטות ומאפיינים מתקדמים של האובייקט Connection**
- **הבנה של Sessions ו-Connection Pooling**

בפרק זה, יוצגו בפניך השיטות (Methods) לעבודה עם מסדי נתונים מתוך דפי ASP שלך. תלמד כיצד להתקשר למסד נתונים, וכיצד לאחסן ולאחזר נתונים. לבסוף, נדון במספר שיטות מתקדמות לעבודה עם connections.

# מילון מושגי גישה לנתונים

כשאני קורא מסמכים טכניים (White Papers) של Microsoft בנושא גישה למסדי נתונים, אני מקבל כאב ראש. ראשי תיבות, ועוד ראשי תיבות, ומעליהם ראשי תיבות נוספים. כדי להבין את אסטרטגיית Microsoft לגישה למסדי נתונים, עליך להבין את משמעותן וחשיבותן של טכנולוגיות כמו MDAC, ADO, ADOX, ADOMD, RDS, OLE DB, ו-ODBC. כדי לחסוך ממך, קורא יקר, כאב ראש דומה, כתבתי את הסעיף הזה, שנועד להבהיר מושגים אלה:

- MDAC - Microsoft Data Access Component - רכיב הגישה לנתונים של Microsoft. רכיב זה מכיל את כל רכיבי התוכנה של Microsoft הקשורים לגישה למסדי נתונים. זהו האוסף הממשי של רכיבי התוכנה שבאפשרותך להוריד מהאתר של Microsoft ולהתקין במחשב שלך. נכון להיום, הגירסה החדשה ביותר היא MDAC 2.5. את MDAC ניתן להוריד מהכתובת http://www.microsoft.com/data, והוא מכיל את רכיבי התוכנה הנידונים בהמשך.
- ADO - ActiveX Data Objects - אובייקטי הנתונים של ActiveX מספקים ממשק ברמת יישום, לגישה לספקי נתונים דוגמת Microsoft SQL Server, או Microsoft Access. באפשרותך להשתמש ב-ADO ישירות מדפי ASP שלך לתקשור (To Communicate) עם מסדי נתונים. ADO הוא הנושא המרכזי של פרק זה והבא אחריו.
- ADOX - הרחבות ADO לאבטחה ולספריות קישור דינמיות (DLL), הכוללות אוסף של אובייקטים המשמשים לניהול אובייקטי מסדי נתונים, ואבטחה. באפשרותך להשתמש באובייקטים אלה מתוך דפי ASP שלך, ליצירת מסדי נתונים וטבלאות, ואחזור המאפיינים שלהם. ADOX כלול גם הוא ב-MDAC.
- ADOMD - ADO Multidimensional - ADO רב מימדי הוא אוסף של אובייקטים המשמשים לעבודה עם נתונים רבי מימדים (Multidimensional Data). בהנחה שאינך עובד עם שרותי Online Analytic Processing (OLAP) הכלולים ב-SQL 7.0, אתה מוזמן מקרב לב להתעלם מאובייקטים אלה.
- RDS - Remote Data Service - שרות הנתונים המרוחקים הוא אוסף של אובייקטים המאפשרים לך לתקשר עם ספקי נתונים הנמצאים על שרתים מרוחקים או בהליכים (Processes) נפרדים. לדוגמה, ניתן להשתמש ברכיב RDS מתוך Internet Explorer לאחזור נתונים ממסד נתונים דרך האינטרנט. ספר זה אינו מכסה את נושא RDS.

- OLE DB - ממשק ברמת-מערכת לספקי נתונים דוגמת Microsoft SQL Server ו- Microsoft Access. לדוגמה, ADO משתמש באוסף הממשקים המסופק על ידי OLE DB, כדי לתקשר עם Microsoft SQL Server. אינך משתמש ב-OLE DB ישירות מדפי ASP שלך. במקום זאת, אתה משתמש ב-ADO כממשק רמה גבוהה ל- OLE DB.
- ODBC - Open Database Connectivity - היו ימים בהם שימש ממשק זה כתקן התקשור למסדי נתונים. כמעט כל מסדי הנתונים הזמינים בשוק הם תואמי ODBC. למרות זאת, Microsoft מחליפה בהדרגה את ODBC ב-OLE DB. נכון להיום, יש ל-Microsoft ספקי OLE DB למסדי נתונים דוגמת SQL, Oracle, ו-Microsoft Jet. Microsoft פיתחה גם ספק OLE DB לשימוש ODBC, במיקרה של מסדי נתונים שאין להם ספקי OLE DB משל עצמם.

באופן כללי, לאחר שהגדרת את החיבור למסד הנתונים שלך, לא תצטרך להשתמש באף אחד מרכיבים אלה, מלבד ADO. ADO מכיל את כל האובייקטים בהם תשתמש באופן שוטף מתוך דפי ASP שלך. לפניך סקירה כללית של שלושת האובייקטים העיקריים הכלולים ב-ADO:

- האובייקט Connection - אובייקט זה מייצג את כל תכונות החיבור למקור הנתונים. עליך לפתוח חיבור בעזרת האובייקט Connection כדי שתוכל לתקשר עם מקור הנתונים. בדרך כלל, נעשה השימוש באובייקט Connection לתקשור עם מסד נתונים דוגמת Microsoft SQL Server או Microsoft Access. עם זאת, ניתן לפתוח connections גם למקורות נתונים אחרים, דוגמת קובץ טקסט פשוט, Microsoft Excel, Microsoft Exchange, Microsoft Index Server ואפילו - באמצעות שימוש ב-ADSI - מערכת הקבצים. בפרק זה, נדון באובייקט Connection בהרחבה.
- האובייקט Recordset - אובייקט זה מייצג את רשומות הנתונים המוחזרים ממקור הנתונים. לדוגמה, כשאתה מציג שאילתה למסד נתונים, רשומות המידע המוחזרות ממסד הנתונים יימצאו באובייקט Recordset. בפרק 13 נדון בפירוט באובייקט זה.
- אובייקט Command - אובייקט זה מייצג פקודה שניתן להפעיל כנגד מקור נתונים. בדרך כלל, משמש האובייקט Command להפעלת שגרות מאוחסנות (Stored Procedures) של SQL. בפרק 14 נדון בפירוט באובייקט זה.

**הערה:**

הטקסט והדוגמאות בפרק זה ובפרקים הבאים יוצאים מתוך הנחה כי הנך משתמש בגרסת ADO הכלולה ב-MDAC 2.1 (ADO גירסה 2.1). באפשרותך לגלות את גרסת ADO המותקנת במחשב שלך על ידי הפעלת התסריט הבא:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Response.Write "ADO Version=" & Con.Version
%>
```

גרסת ADO החדשה ביותר ניתנת להורדה ב:
http://www.microsoft.com/data

**הערת העורך:**

מסד נתונים מכיל רשומות. כל **רשומה** מכילה **שדות**. מכיון שאנו מדברים על מסדי נתונים טבלאיים, מתייחסים לרשומה בתור **שורה** ולשדה בתור **עמודה**.

# יצירת Connections עם OLE DB ו-ODBC

לגישה למסד נתונים מתוך דפי ASP, יש להשתמש ב-ADO. ADO מכיל את אוסף האובייקטים בהם תשתמש בתסריטים שלך כדי ליצור קישורים למסד נתונים וקריאת רשומות מטבלת מסד הנתונים.

אבל, ADO אינו מתקשר עם מקור הנתונים ישירות, אלא דרך ממשק ביניים הנקרא OLE DB. OLE DB מספק ממשק ברמת מערכת למקורות נתונים, דוגמת Microsoft SQL Server ו-Microsoft Access.

באופן כללי, לאחר יצירת החיבור למסד הנתונים, ניתן להתעלם מקיומו של OLE DB, כיון שכל עבודתו של ממשק זה מתבצעת ברקע. לעולם לא תיגש ל-OLE DB ישירות מתוך דפי ASP שלך. ביצירת החיבור הראשוני למסד הנתונים, לעומת זאת, עליך להיות מודע לקיומו של OLE DB, כיון שביכולתך לבחור לגשת למסד נתונים באמצעות ספקי OLE DB שונים.

קיימות שתי דרכים בהן יכול ספק OLE DB לספק גישה למסד נתונים: באופן בלתי ישיר, דרך התקן ODBC, או ישירות, במידה והוא ספק הנכתב במטרה לתמוך ב-OLE DB. מדוע הדבר כה מורכב?

הדרך המסורתית לתקשור עם מסד נתונים הייתה באמצעות ממשק ODBC (חיבוריות פתוחה למסד נתונים - Open Database Connectivity). ODBC מספק תקן נפוץ לתקשור עם מסדי נתונים, תוך שימוש בשפת SQL. בכל פעם שספק (Vendor) מפתח גירסה חדשה של מסד נתונים, הוא נוהג לספק עמה גם אוסף חדש של התקני ODBC, כך שיישומים קיימים יוכלו לתקשר עם מסד הנתונים החדש. תקן ODBC הקל על חייהם של מפתחי מסדי נתונים, על ידי יצירת תקן אחיד.

**הערה:**

SQL הוא קיצור של Structured Query Language, כלומר "שפת שאילתות מבנית". שפה זו היא תקן התקשור עם מסדי נתונים. ראה נספח D, לרשימה מהירה של פקודות SQL החשובות ביותר והתחביר שלהן.

אבל, Microsoft מחליפה בהדרגה את תקן ODBC ב-OLE DB. הבעיה עם ODBC היא כי הוא תוכנן למטרה הצרה של עבודה עם מסדי נתונים התומכים בשפת SQL בלבד. לעומתו, תוכנן OLE DB לעבוד עם תחום רחב בהרבה של מקורות נתונים. לדוגמה, Microsoft פיתחה ספקי OLE DB למקורות נתונים מגוונים, דוגמת Microsoft Exchange, Microsoft Index Server ו- ADSI.

כדי להאיץ את אימוצו של OLE DB, יצרה Microsoft ספק OLE DB הפועל עם התקני ODBC מסורתיים. ספק OLE DB להתקני ODBC ממפה את שיטות וממשקי OLE DB ל-API של ODBC. על ידי שימוש בספק זה, ניתן להשתמש ב-OLE DB לתקשור עם מסדי נתונים להם אין עדיין ספק OLE DB משל עצמם.

כך, קיימות למעשה שתי שיטות לשימוש ב-ADO ליצירת חיבור למסד נתונים. מצד אחד, ניתן ליצור חיבור על ידי שימוש בספק OLE DB להתקני ODBC. קישור מסוג זה יעבוד עם כל מסד נתונים עם התקן ODBC (כלומר, כמעט כל מסדי הנתונים הקיימים בשוק). מצד שני, ניתן ליצור חיבור על ידי שימוש בספק OLE DB טבעי (Native OLE DB Provider).

באיזה ספק לבחור? כשהדבר אפשרי, רצוי להשתמש בספקי OLE DB טבעיים, המספקים גישה יעילה יותר לנתונים. Microsoft, ומספר חברות נוספות, פיתחו ספקי OLE DB טבעיים לרבים ממסדי הנתונים הנפוצים ביותר, וביניהם Microsoft SQL Server, Oracle ו-Microsoft Access. השתמש ב-ODBC רק כאשר לא קיים ספק OLE DB טבעי למקור הנתונים שלך. הסעיפים הבאים מספקים דוגמאות לאופן השימוש בשני סוגי החיבורים, בחיבור ל-Microsoft SQL Server ול-Microsoft Access.

# חיבור ל-Microsoft SQL Server

לפני שתוכל לתקשר עם מסד נתונים דוגמת Microsoft SQL Server, עליך לפתוח אליו חיבור (connection). הדבר מתבצע על ידי פתיחת האובייקט Connection של ADO עם מחרוזת חיבור (Connection String).

מחרוזת חיבור (Connection String) כוללת מידע של דוגמת מיקום מקור הנתונים, ואמצעי האימות (מבחינת אבטחה) של המשתמש הפותח את החיבור (connection). קיימות שתי שיטות עיקריות להעברת מידע זה. ניתן לספק את המידע כולו במחרוזת החיבור, או להשתמש במחרוזת החיבור כמראה מקום לקובץ חיצוני (או לרשומה ברישום המערכת, System Registry) המכיל מידע זה.

# חיבור ל-SQL Server באמצעות OLE DB

השיטה המועדפת לחיבור למסד נתונים מסוג Microsoft SQL Server היא על ידי שימוש בספק OLE DB טבעי לשרת ה-SQL. הדוגמה הבאה פותחת connection לשרת Microsoft SQL בשם yourServer :

```
<%
Set Con  =  Server.CreateObject( "ADODB.Connection" )
Set.Open "PROVIDER=SQLOLEDB; DATA  SOURCE=yourServer;
    ↳ UID=sa;PWD=secret;DATABASE=Pubs "
%>
```

תסריט זה יוצר מופע של האובייקט Connection של ADO. לאחר מכן, נפתח connection לשרת ה-SQL, עם מחרוזת חיבור הכוללת חמישה פרמטרים : ספק OLE DB, מקור המידע, מסד הנתונים, זיהוי המשתמש והסיסמה.

הפרמטר PROVIDER משמש לציון שמו של ספק OLE DB שישמש לחיבור. דוגמה זו משתמשת בספק OLE DB טבעי לשרת ה-SQL. אם לא מסופק פרמטר זה, ברירת המחדל היא שימוש בספק OLE DB להתקני ODBC. לגישה מהירה יותר למסד הנתונים, יש לציין פרמטר זה, המציין את ספק OLE DB לשרת ה-SQL.

הפרמטר DATA SOURCE משמש לציון שם שרת ה-SQL שלך. לדוגמה, אם שרת ה-SQL מותקן במחשב ששמו Plato, עליך לספק לפרמטר זה את הערך Plato. אם שרת מסד הנתונים מותקן באותו המחשב בו מותקן שרת האינטרנט, ניתן לספק לפרמטר זה את הערך LocalServer.

הפרמטר UID מציין את שם ההתחברות לשרת ה-SQL שייעשה בו שימוש בחיבור זה. דוגמה זו משתמשת בשם sa לחיבור למסד הנתונים. בדרך כלל, לעומת זאת, עדיף להשתמש בשם אחר, עם הרשאות אבטחה מוגבלות יותר.

הפרמטר UID הוא אופציונלי. אם לא תציין אותו, ייעשה חיבור באמצעות החשבון IUSER_MachineName (כאשר MachineName הוא שם המחשב ממנו מתבצע ניסיון ההתחברות). אם מסד הנתונים אינו כולל משתמש בשם זה, החיבור ייכשל.

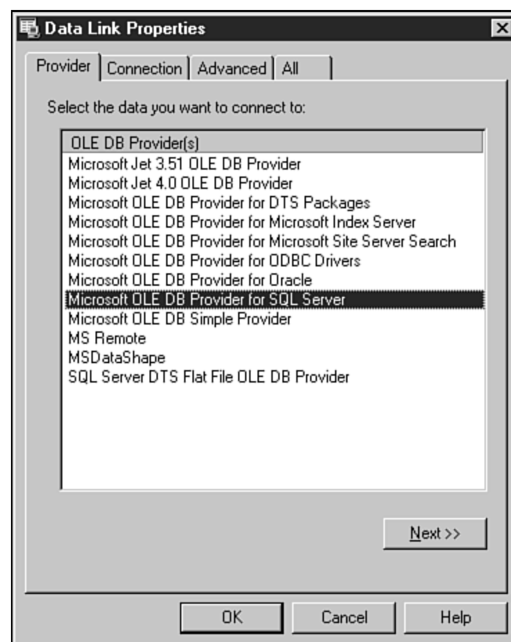
הפרמטר PWD מכיל את סיסמת ההתחברות ל-SQL. ניתן לקבוע סיסמה זו ב-SQL Enterprise Manager. החיבור למסד הנתונים ייכשל אם לא תסופק הסיסמה הנכונה.

הפרמטר DATABASE משמש לציון מסד נתונים ספציפי הממוקם בשרת מסד הנתונים שלך. גם פרמטר זה אופציונלי. אם לא יצוין מסד נתונים ספציפי, ייעשה שימוש במסד ברירת המחדל (ניתן לקבוע את מסד ברירת המחדל ב-SQL Enterprise Manager).

במקום ליצור את מחרוזת החיבור באופן ידני, ניתן להשתמש ב-Microsoft Data Link כדי ליצור באופן אוטומטי קובץ, המכיל את מידע החיבור. לאחר יצירת קובץ זה, ניתן לפנות אליו מתוך מחרוזת החיבור.

כדי ליצור קישור נתונים (Data Link) חדש, יש לבצע את הצעדים הבאים:

1. לחץ לחיצה ימנית על שולחן העבודה של Windows, הצבע על New, ובחר Microsoft Data Link. כאן ניתן לקבוע שם לקישור הנתונים החדש (למשל, myDataLink).
2. לחץ לחיצה כפולה על קישור הנתונים החדש. הדבר יפעיל את יישומון לוח הבקרה Data Link.
3. לאחר מכן, בחר ספק נתונים, על ידי בחירה ב-Microsoft OLE DB Provider for SQL Server מהתיקיה Provider (ראה תרשים 12.1). לחץ על Next.




**תרשים 12.1** יצירת Data Link חדש לשרת SQL

4. בתיקיה Connection, הקלד את שם השרת שלך, ספק את נתוני האימות, ובחר את מסד הנתונים המסוים אליו ברצונך לגשת.

5. לחץ על OK לשמירת קישור הנתונים החדש שלך.

לאחר יצירת קישור הנתונים, תוכל לפתוח connection למסד נתונים באמצעות תסריט המוצג כדוגמה להלן:

```
<%
Set Con = Server.CreateObject("ADODB.Connection")
Set.Open "File Name=C:\myDataLink.UDL;UID=sa;PWD=secret"
%>
```

תסריט זה מניח כי יצרת קישור נתונים בשם myDataLink ושמרת אותו בתיקיית השורש של כונן C.

אחד היתרונות בשימוש בקישור נתונים הוא בכך שניתן לשמור את שם המשתמש והסיסמה יחד עם הקובץ. כדי לעשות זאת, בחר באפשרות Allow Saving Password. כשהסיסמה נשמרת באופן זה, ניתן לפתוח את ה-connection באמצעות התסריט:

```
<%
Set Con = Server.CreateObject("ADODB.Connection")
Set.Open "File Name=C:\myDataLink.UDL"
%>
```

**הערה:**

שים לב כי מרבית הדוגמאות בספר זה עושות שימוש בקישור נתונים המכיל את שם המשתמש והסיסמה, לפתיחת connection למסד נתונים.

# חיבור לשרת SQL עם ODBC

הדרך המסורתית לחיבור לשרת מסד נתונים של SQL היא באמצעות ODBC. כשאתה פותח connection מבלי לציין במפורש ספק OLE DB, החיבור עושה שימוש בספק OLE DB להתקני ODBC.

קיימות שלוש שיטות לפתיחת connection לספק זה. כל שיטה מחפשת את נתוני ה-connection במיקום אחר. המידע יכול להיות מאוחסן ברישום המערכת (Registry), בקובץ טקסט, או במחרוזת החיבור (string connection).

## יצירת System DSN לשרת SQL של Microsoft

כדי לאחסן את מידע ה-connection ברישום המערכת, יש ליצור System Data Source Name (DSN). לפניך דוגמה לתסריט הפותח connection למסד נתונים באמצעות System DSN:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "DSN=MyDSN;UID=sa;PWD=secret;DATABASE=Pubs"
%>
```

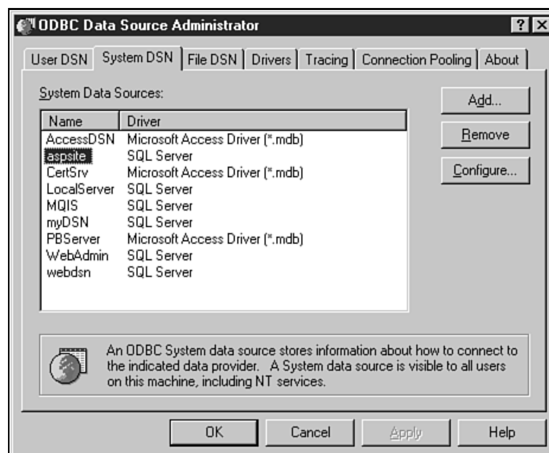
תסריט זה יוצר מופע של האובייקט Connection של ADO באמצעות השיטה CreateObject. לאחר מכן, נפתח האובייקט Connection על ידי העברת מחרוזת חיבור המכילה ארבעה פרמטרים: System DSN, מזהה משתמש, סיסמה, ושם מסד נתונים.

על מזהה המשתמש (User ID) והסיסמה להתאים לשם משתמש ולסיסמה ברי תוקף בשרת SQL. ניתן להשתמש בכל שם משתמש שיצרת עבור מסד הנתונים, אך כדאי לבדוק האם למשתמש יש הרשאות מתאימות לגישה למסד הנתונים הרצוי לך.

פרמטר מסד הנתונים הינו אופציונלי. אם לא תכלול אותו, יפתח חיבור (connection) למסד ברירת המחדל המוגדר עבור שם המשתמש.

לפני הפעלת תסריט זה, יש ליצור System DSN ברישום המערכת. לפניך השלבים ליצירת System DSN.

1. הפעל את ODBC Data Source Administrator. יישומון זה ממוקם בלוח הבקרה (Control Panel), תחת הכותרת ODBC Data Sources (ראה תרשים 12.2).




**תרשים 12.2** ODBC Date Source Administrator

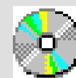
2. מתוך ODBC Data Source Administrator, בחר בכרטיסיה System DSN.
3. לחץ על Add, בחר את התקן שרת ה-SQL, ולחץ על Finish.
4. הקלד שם (לדוגמה, MyDSN) ותאור (Description), ולאחר מכן בחר את השרת אליו ברצונך להתחבר. אם שרת ה-SQL ו-IIS מותקנים באותו מחשב, בחר Local. אם השרת נמצא במקום זה או אחר באינטרנט, באפשרותך לציין כתובת IP או Domain Name. לחץ על Next.

5. בחר בשיטה לאימות (Authentication). נתוני האימות באים לידי שימוש בעת יצירת ה-System DSN בלבד. לא נעשה שימוש בנתונים אלה בעת השימוש ב-System DSN בדפי ASP. לחץ על Next שלוש פעמים, כדי לדלג על שני המסכים הבאים.

6. לבסוף, ניתנת לך אפשרות לבדיקת החיבור החדש. אם הכל פועל כמתוכנן, יצרת לעצמך System DSN חדש.

יצירת System DSN מתבצעת פעם אחת בלבד. לאחר יצירתו, ניתן להשתמש בו מתוך כל אחד מתסריטי ASP שבאותו המחשב. במידה ותעביר את אתר האינטרנט שלך לשרת אחר, או שתשנה את מיקומו של שרת מסד הנתונים שלך, יהיה עליך ליצור System DSN נוסף, בעל אותו השם.

**הערה:**



ניתן לפתוח את ההתקשרות למסד נתונים בצורה פשוטה יותר מבלי לרשום את השם ואחר כך את הערך, אלא רק את הערך. לדוגמה במקום לרשום:

```
con.open "DSN=myDSN;UID=Sa"
```

ניתן לרשום:

```
Con.open "myDSN"_"Sa"
```

# יצירת File DSN לשרת SQL של Microsoft

במקום לאחסן את מידע ה-connection ברישום המערכת, ניתן לאחסנו בקובץ טקסט, על ידי יצירת File DSN. ליצירת File DSN, הפעל את ODBC Date Source Administrator ובחר בכרטיסיה File DSN. לחץ על Add, ותינתן לך האפשרות לציין שם ומיקום לקובץ המיועד לאחסון מידע החיבור. השלבים הבאים זהים לשלבי יצירת ה-System DSN.

בהנחה ששם ה-File DSN שלך הוא myFileDSN, תוכל לפתוח חיבור (connection) למסד נתונים באמצעות התסריט הבא:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILEDSN=myFileDSN;UID=sa;PWD=secret;DATABASE=Pubs"
%>
```

שים לב שהתסריט זהה לזה שבדוגמה הקודמת, למעט השימוש ב-FILEDSN במקום ב-DSN.

# יצירת DSN-less Connection לשרת SQL של Microsoft

אם ברצונך לכלול את כל מידע החיבור (connection) בתוך מחרוזת חיבור, באפשרותך לפתוח DSN-Less Connection. במקרה כזה, אין צורך לקבוע דבר ב-ODBC Data Source Administrator. תסריט זה שקול לשני התסריטים שבדוגמאות הקודמות:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "DRIVER={SQL Server};SERVER=yourServer;UID=sa;PWD=secret"
%>
```

במקום השימוש בפרמטר DSN או FILEDSN, משתמשת מחרוזת חיבור זו בפרמטרים DRIVER ו-SERVER. כדי שתוכל להשתמש בתסריט זה, יהיה עליך להחליף את הביטוי yourServer בשם שרת מסד הנתונים שלך. צורת חיבור זו מהווה יתרון במידה ופיתחת אתר עבור גוף מסוים, בחברה שלך ועל השרת שלך שעליו יושב בסיס הנתונים, כאשר תצטרך להעביר את האתר לשרתים של אותו גוף, לא תצטרך להגדיר מחדש את ה-DSN.

# פתיחת Database Connection באינטרנט

תוכל להשתמש בכל אחת משיטות החיבור שהוצגו, כדי לתקשר עם שרתי מסד נתונים של SQL, ללא תלות במיקומם. לדוגמה, ניתן לאחזר את כל הרשומות (Records) מטבלת מסד נתונים הממוקמת באנגליה מתוך דף ASP המופעל במחשב בקליפורניה (או בהרצליה פיתוח). אם ברצונך לתקשר עם שרת מסד נתונים על גבי האינטרנט, עליך לקבוע את תצורת השרת והלקוח כך שישתמשו בפרוטוקול TCP/IP.

כדי לקבוע את תצורת שרת האינטרנט שלך לשימוש בפרוטוקול TCP/IP, הפעל את Client Network Utility מקבוצת היישומים Microsoft SQL Server שבתפריט Programs. לאחר מכן, אפשר (Enable) את ספריית הרשת של TCP/IP (TCP/IP Network Library). אם Client Network Utility אינו מותקן בשרת האינטרנט שלך, באפשרותך להתקינו על ידי הפעלת תוכנית ההתקנה שלו, שבתקליטור ההתקנה של שרת ה-SQL.

כדי לקבוע את תצורת שרת מסד הנתונים SQL Server גירסה 7.0 שלך לשימוש בפרוטוקול TCP/IP, השתמש ב-Server Network Utility הנמצא בקבוצת היישומים Microsoft SQL Server 7.0 שבתפריט Programs (ספריית ה-TCP/IP פעילה כברירת מחדל). כדי לאפשר לשרת מסד הנתונים SQL גירסה 6.5 להשתמש ב-TCP/IP, הפעל את תוכנת ההתקנה מתוך קבוצת היישומים Microsoft SQL Server 6.5 שבתפריט Programs.

בהנחה ששרת מסד הנתונים שלך ממוקם ב-host.domain.com, תוכל להתחבר אליו באמצעות התסריט הבא:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "PROVIDER=SQLOLEDB;DATA SOURCE=host.domain.com;
    ↳ UID=sa;PWD=secret;DATABASE=Pubs"
%>
```

לאחר קביעת התצורה של שרת ולקוח מסד הנתונים לשימוש ב-TCP/IP, ניתן גם להשתמש בכל שירותי הלקוח (Client Utilities) של SQL באינטרנט. לדוגמה, תוכל להשתמש ב-Enterprise Manager וב-Query Analyzer לניהול מרחוק של מסדי הנתונים של שרת ה-SQL שלך.

# חיבור למסד הנתונים Microsoft Access

אם אתה צופה תנועת מבקרים רבה באתר האינטרנט שלך, רצוי כי לא תשתמש ב-Microsoft Access כמסד הנתונים של האתר. מערכת מסד נתונים זו היא מערכת שולחנית, ולא מערכת שרת/לקוח. משמעות הדבר היא כי המערכת לא תוכננה לתמוך בו זמנית במספר משתמשים העולה על מספר המבקרים הצפויים באתר אינטרנט צנוע. לדוגמה, Microsoft Access אינה כוללת תמיכה בנעילה רבת משתמשים מתקדמת (Advanced MultiUser Locking), יומני תנועות (Transaction Logs) או ניהול אינדקסים יעיל (Efficient Indexing).

**הערה:**

אלא אם צוין במפורש, כל הדוגמאות בספר מתייחסות לעבודה עם מסד נתונים מסוג Microsoft SQL Server.

למרות אזהרה זו, סעיף זה מסביר כיצד לפתוח connection למסד נתונים מסוג Access מתוך דפי ASP שלך. תלמד ליצור connection הן על ידי שימוש בספק OLE DB טבעי של Access, והן על ידי שימוש ב-ODBC. לבסוף, תלמד כיצד ליצור, מתוך דפי ASP שלך, מסד נתונים Access חדש, על ידי שימוש ב-ADOX.

## יצירת Connection ל-Microsoft Access על ידי שימוש ב-OLE DB

השיטה המועדפת לחיבור למסד נתונים מסוג Microsoft Access היא על ידי שימוש בספק OLE DB טבעי. קיימים שני ספקי OLE DB טבעיים המיועדים לשימוש עם Microsoft Access: MICROSOFT.JET.OLEDB.3.5 ו-MICROSOFT.JET.OLEDB.4.0. כל אחד מספקים אלה מותאם לגרסאות שונות של מסד הנתונים Access. לפניך דוגמה לפתיחת חיבור תוך שימוש ב- OLEDB.4.0:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "PROVIDER=MICROSOFT.JET.OLEDB.4.0;
   ↳ DATA SOURCE=c:\myAccess.mdb"
%>
```

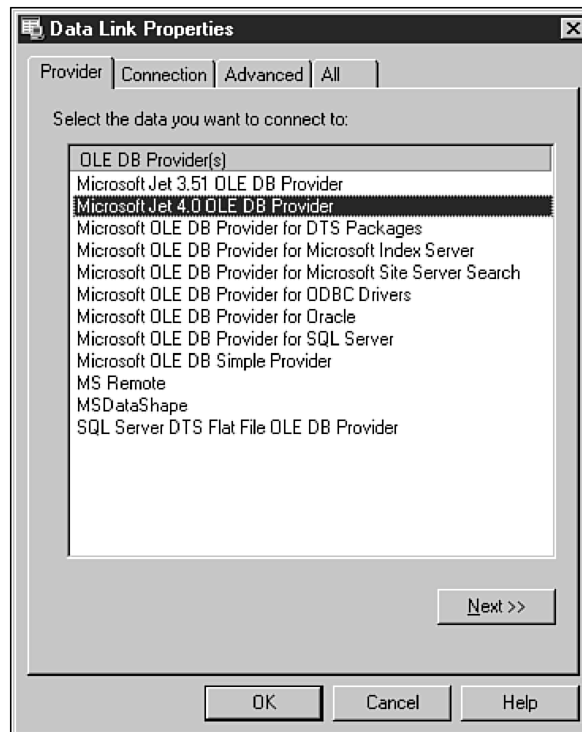
שים לב כי הפרמטר DATA SOURCE מספק נתיב למסד הנתונים של Access (קובץ ה-mdb).

במקום לספק את כל מידע החיבור (connection) במחרוזת החיבור, ניתן לאחסן את המידע בקישור נתונים (Data Link). לפניך השלבים ליצירת קישור נתונים של Access:

1. לחץ לחיצה ימנית על שולחן העבודה ובחר ב-New, ולאחר מכן בחר ב-Microsoft Data Link. בשלב זה יהיה עליך לתת שם לקובץ קישור הנתונים שלך (לדוגמה, myDataLink).

2. לחץ לחיצה כפולה על קישור הנתונים החדש. פעולה זו תפעיל את יישומון לוח הבקרה Data Link.

3. בחר בספק המידע, על ידי בחירה ב-Microsoft Jet OLE DB Provider בכרטיסיה Provider (ראה תרשים 12.3).




**תרשים 12.3** יצירת קישור נתונים ל-Microsoft Access

4. בכרטיסיה Connection, הקלד את נתיב מסד הנתונים של Access (קובץ ה-mdb).

5. לחץ על OK, כדי לשמור את קישור הנתונים החדש.

לאחר יצירת קישור הנתונים החדש, ניתן להתייחס אליו במחרוזת חיבור, כך:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "File Name=C:\myDataLink.UDL"
%>
```

## יצירת Connection ל-Microsoft Access על ידי שימוש ב-ODBC

אם תעדיף ליצור חיבור (connection) למסד נתונים של Access על ידי שימוש ב-ODBC, יעמדו בפניך שלוש אפשרויות. תוכל לאחסן את מידע החיבור ברישום המערכת, בקובץ טקסט, או במחרוזת חיבור.

## יצירת System DSN ל-Microsoft Access

כדי לאחסן את מידע החיבור ברישום המערכת, יש ליצור System DSN. לפניך השלבים ליצירת Access System DSN:

1. הפעל את ODBC Data Source Administrator. יישומון זה ממוקם בלוח הבקרה (Control Panel), תחת הכותרת ODBC Data Sources.
2. מתוך ODBC Data Source Administrator, בחר בכרטיסיה System DSN.
3. לחץ על Add, בחר את התקן ה-Access, ולחץ על Finish.
4. הקלד שם (לדוגמה, AccessDSN) ותיאור (Description) לחיבור. תוצג בפניך האפשרות ליצור מסד נתונים חדש של Access, או שתוכל ללחוץ על Select, כדי לבחור מסד נתונים של Access שכבר קיים בשרת שלך. אם תלחץ על Advance, יוצגו בפניך מספר אפשרויות מתקדמות. לדוגמה, תוכל לקבוע תיקיית ברירת מחדל לקבצי מסד הנתונים של Access, או לציין ברירת מחדל לשם המשתמש והסיסמה. לחץ על OK כדי לסיים.

לאחר יצירת ה-System DSN, ניתן להשתמש בו מתוך כל אחד מדפי ASP הממוקמים באותו מחשב. לדוגמה, התסריט הבא פותח connection למסד נתונים של Access תוך שימוש ב-System DSN בשם AccessDSN:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "DSN=AccessDSN"
%>
```

תסריט זה יוצר מופע של האובייקט Connection של ADO, ופותח אותו באמצעות מחרוזת חיבור (string connection). ניתן לכלול בתסריט גם מידע אימות, על ידי אספקת הפרמטרים UID ו-PWD, כך:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "DSN=AccessDSN;UID=admin;PWD=secret"
%>
```

## יצירת File DSN ל-Microsoft Access

התהליך ליצירת File DSN דומה לתהליך יצירת System DSN. כדי ליצור File DSN הפעל את ODBC Data Source Administrator, ובחר בכרטיסיה File DSN. לחץ על Add, ולאחר מכן ציין את שם ומיקום הקובץ בו יישמר מידע החיבור (connection). השלבים הבאים זהים לשלבי יצירת ה-System DSN.

לאחר יצירת File DSN, ניתן להשתמש בו לפתיחת חיבור (connection) למסד נתונים של Access באמצעות התסריט הבא:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILEDSN=AccessDSN"
%>
```

תסריט זה פותח חיבור (connection) על ידי כך שהוא מכיל את שם ה-File DSN במחרוזת החיבור. ניתן לספק גם מידע אימות, על ידי שימוש בפרמטרים UID ו-PWD.

## יצירת DSN-less Connection ל-Microsoft Access

לבסוף, אם אינך מעוניין ליצור System DSN או File DSN, ניתן לפתוח DSN-Less Connection. כדי ליצור DSN-Less Connection למסד נתונים של Access, יש לכלול שני פרמטרים נוספים במחרוזת החיבור - הפרמטרים DRIVER ו-DBQ:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=C:\myDatabase.mdb"
%>
```

הפרמטר DRIVER משמש לציון התקן Microsoft Access. הפרמטר DBQ מספק את הנתיב לקובץ מסד הנתונים של Access (קובץ ה-mdb). בדוגמה שלנו, הקובץ ממוקם בתיקיית השורש של כונן :C.

## יצירת מסד נתונים של Access מתוך דף ASP

אין צורך לרכוש את חבילת Microsoft Access כדי להשתמש במסד נתונים של Access באתר אינטרנט. ניתן ליצור מסד נתונים Access חדש, על ידי שימוש ב-ODBC Data Source Administrator, ביישומון Data Link, או באופן תכנותי, מתוך תסריט ASP.

ADO עצמו אינו כולל שיטות ליצירת מסד נתונים Access חדש. כדי ליצור מסד נתונים חדש מתוך דף ASP, יש להשתמש ב-ADOX (הרחבות ה-ADO לתמיכה בספריות קישור דינמיות (DLL) ובאבטחה). ADOX כלול בהתקנת MDAC 2.5 שניתן להוריד מהאתר של Microsoft (http://www.microsoft.com/data). התסריט הבא יוצר מסד נתונים Access חדש, ופותח אליו connection. ניתן למצוא את התסריט בתקליטור המצורף, בקובץ newAccess.asp.

```
<%
DBName = "c:\myAccessDB.mdb"
Set myCat = Server.CreateObject( "ADOX.Catalog" )
myCat.Create "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & DBName
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "PROVIDER=MICROSOFT.JET.OLEDB.4.0;DATA SOURCE=" & DBName
%>
```

תסריט זה יוצר מופע של האובייקט Catalog של ADOX, ומשתמש בשיטה Create של האובייקט כדי ליצור מסד נתונים Access חדש. לאחר מכן, נפתח connection למסד הנתונים החדש על ידי שימוש בספק ODBC טבעי.

## סגירת Connection פתוח

אין צורך לסגור חיבור (connection) פתוח למסד נתונים, כשמשתמשים בדפי ASP. חיבור פתוח ייסגר באופן אוטומטי בסיום עיבוד דף ה-ASP.

עם זאת, נהוג לסגור חיבור (connection) באופן מפורש לאחר סיום השימוש בו. מסד נתונים מסוגל לטפל במספר מוגבל של חיבורים (connections) בזמן נתון. על ידי סגירה של חיבורים באופן מפורש, מתפנים משאבים בשרת מסד הנתונים.

לסגירה באופן מפורש של חיבור (connection), יש להשתמש בשיטה Close של האובייקט Connection. ראה להלן:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=C:\myDataLink.UDL"
'...Do Some Database Work
Con.Close
Set Con = NOTHING
%>
```

דוגמה זו פותחת חיבור למסד נתונים, וסוגרת אותו בתום השימוש, באמצעות השיטה Close. המשפט האחרון בתסריט מפנה את האובייקט Connection מזיכרון המחשב.

# הפעלת משפטי SQL באמצעות האובייקט Connection

לאחר שפתחת חיבור (connection) למסד נתונים, תוכל אפילו להשתמש בו לדבר מה מועיל. באמצעות השיטה Execute של האובייקט Connection, ניתן להפעיל כל משפט SQL. בסעיפים הבאים נלמד על ידי דוגמאות כיצד ליצור טבלת מסד נתונים חדשה תוך שימוש ב-Microsoft SQL Server. תלמד כיצד להוסיף, לעדכן, למחוק ולבחור נתונים מתוך טבלת מסד נתונים.

# יצירת טבלת מסד נתונים חדשה

התסריט הבא יוצר טבלת מסד נתונים חדשה בשם Webusers, במסד הנתונים של ברירת המחדל של שם המשתמש sa.

```
<%
MySQL="Create Table Webusers(username, VARCHAR(20), password, VARCHAR(20))"
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Con.Execute MySQL
%>
```

המשפט Create Table של SQL משמש ליצירת טבלה חדשה. הטבלה נוצרת עם שתי עמודות: username ו-password. שתי העמודות הן עמודות מסוג VARCHAR, בעלות אורך מקסימלי של 20 תווים.

**הערה:**

כדי להסיר טבלה לאחר יצירתה, ניתן להשתמש במשפט Drop Table של SQL.
המשפט הבא מסיר את הטבלה Webusers:

```
Drop Table Webusers
```

# הוספת נתונים לטבלת מסד נתונים

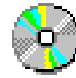
המשפט Insert של SQL משמש להוספת רשומה חדשה של נתונים לטבלת מסד נתונים.
להלן דוגמה:

```
<%
MySQL="Insert Webusers ( username, password )
    ↳ values ( 'Andrew Joness', 'secret' )"
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Con.Execute MySQL
%>
```

תסריט זה מוסיף את שם המשתמש Andrew Jones ואת הסיסמה secret לתוך הטבלה Webusers. שים לב לשימוש בגרשים בודדים (Single Quotes) מסביב למחרוזות שהוספו.

**הערה:**

בעת הוספה של נתונים למסד נתונים Access, יש להשתמש ב- Insert Into ולא ב- Insert.



משפט SQL המופעל הוא בסך הכל מחרוזת. באפשרותך לטפל במחרוזת זו בכל דרך שתחפוץ, כדי להוסיף ערכים שונים. תדפיס 12.1 מדגים כיצד ניתן לאחזר נתונים מטופס HTML ולהוסיף אותם לטבלה Webusers (ראה תרשים 12.4).

**תרשים 12.4** הוספת נתונים דרך טופס HTML

**תדפיס 12.1** insertData.asp

```
<%
FUNCTION fixQuotes( theString )
    fixQuotes = Replace( theString, " ' ", " ' ' " )
END FUNCTION
username = TRIM( Request( "username" ) )
password = TRIM( Request( "password" ) )
IF username <> "" AND password <> "" THEN
    mySQL = "INSERT Webusers (username, password ) VALUES "
    mySQL = MySQL & "( ' " & fixQuotes( username ) & " ' , ' " &
        ↳ fixQuotes( password ) & " ' )"
    Set Con = Server.CreateObject( "ADODB.Connection" )
    Con.Open "FILE NAME=c:\myDataLink.UDL"
    Con.Execute mySQL
END IF
%>
<HTML>
<HEAD><TITLE>Insert Data</TITLE>
<BODY>
<FORM method="post" action="insertData.asp">
<P>Username: <input name="username" size=20>
```

```
<P>Password: <input name="password" size=20>
<P><input type="submit" value="Add Data">
</FORM>
</BODY>
</HTML>
```

דף ASP זה מאחזר ערכים משני שדות בטופס HTML: username ו-password. לאחר מכן, משמשים הערכים להרכבת משפט Insert תקין של SQL. בעת הקריאה לשיטה Execute של האובייקט Connection, מוספים שם המשתמש והסיסמה לטבלה Webusers.

שים לב כי דף ASP זה מכיל פונקציה בשם ()fixQuotes. פונקציה זו מחליפה כל מופע של תו הגרש (') בשני תווי גרש. ללא שימוש בפונקציה זו, במקרה שמשתמש מסוים ינסה להוסיף שם או סיסמה המכילים גרש הוא יחולל שגיאה, כיון ששפת SQL משתמשת בתו הגרש כדי לסמן התחלה וסיום של ערך מסוג מחרוזת. יש להשתמש בפונקציה ()fixQuotes בכל מקרה בו צריך להוסיף נתונים למסד נתונים על ידי הפעלת משפט SQL באמצעות האובייקט Connection.

# עדכון נתונים בטבלת מסד נתונים

כדי לשנות נתונים קיימים בטבלת מסד נתונים, יש להשתמש במשפט Update של SQL. ניתן להשתמש במשפט Update לשינוי כל רשומות הטבלה, או רשומה ספציפית בלבד. לפניך דוגמה המשנה את כל הסיסמאות בטבלה Webusers:

```
<%
MySQL="Update Webusers SET password = 'apple' "
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Con.Execute MySQL
%>
```

תסריט זה משנה את כל הסיסמאות ל-apple. בדרך כלל, לא תרצה לשנות את כל הרשומות בטבלת מסד הנתונים. כדי לשנות רשומות ספציפיות בלבד, ניתן לכלול סעיף WHERE במשפט ה-SQL:

```
<%
MySQL="Update Webusers SET password = 'apple'
   ↳ WHERE username ='Andrew Jones' "
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Con.Execute MySQL
%>
```

תסריט זה משנה את הסיסמה רק במידה ושם המשתמש שווה ל-Andrew Jones. כברירת מחדל, שרת ה-SQL אינו תלוי רישיות (Not Case Sensitive). משמעות הדבר היא כי התסריט ישנה את הסיסמאות גם למשתמשים ששם המשתמש שלהם הוא andrew jones, ANDREW JONES או AnDrEw JoNeS.

# מחיקת נתונים מטבלת מסד נתונים

כדי להסיר רשומה מטבלת מסד נתונים, יש להשתמש במשפט Delete של SQL. התסריט הבא ימחק את כל הרשומות מהטבלה Webusers :

```
<%
MySQL="Delete Webusers"
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Con.Execute MySQL
%>
```

תסריט זה מסיר את כל הרשומות מהטבלה Webusers. עם זאת, הוא אינו מסיר את הטבלה עצמה.

**הערה:**

בדרך כלל, לא כדאי להשתמש במשפט Delete למחיקת כל הרשומות מטבלת מסד נתונים, היות והדבר מעמיס עבודה רבה מדי על שרת ה-SQL (וממלא את יומן התנועות (transaction)). כדי להסיר את כל הרשומות בטבלה, יש להשתמש במשפט Truncate Table, כך:

Truncate Table Webusers

כדי להסיר רשומה ספציפית בלבד מטבלת מסד נתונים, ניתן להשתמש בסעיף WHERE במשפט ה-SQL. התסריט הבא יסיר רק את הרשומות בהן הסיסמה שווה ל-secret :

```
<%
MySQL="DELETE Webusers WHERE password = 'secret' "
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Con.Execute MySQL
%>
```

# בחירת נתונים מתוך טבלת מסד נתונים

כדי לאחזר רשומות מטבלת מסד נתונים, יש להשתמש במשפט Select של SQL. לדוגמה, לאחזור כל הרשומות מהטבלה Webusers, ניתן להשתמש בתסריט הבא:

```
<%
MySQL="SELECT * FROM Webusers"
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Set RS = Con.Execute( MySQL )
While Not RS.EOF
    Response.Write RS( "username" ) & " = " & RS( "password" ) & "<br>"
    RS.MoveNext
WEND
%>
```

תסריט זה מאחזר את כל עמודות ושורות הטבלה Webusers לתוך אובייקט Recordset בשם RS. לאחר מכן, מוצגות כל רשומות האובייקט Recordset לדפדפן, בתוך לולאת WHILE...WEND.

**הערה:**

בפרק 13 תלמד בהרחבה אודות האובייקט Recordset.

אם ברצונך לאחזר מטבלת מסד נתונים עמודות ספציפיות בלבד, תוכל לציין את העמודות שברצונך לאחזר בהצהרה Select. לפניך דוגמה המאחזרת רק את הערכים מעמודת הסיסמאות בטבלה Webusers:

```
<%
MySQL="SELECT password FROM Webusers"
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Set RS = Con.Execute( MySQL )
While Not RS.EOF
    Response.Write RS( "username" ) & " = " & RS( "password" ) & "<br>"
    RS.MoveNext
WEND
%>
```

כדי לבחור רשומות ספציפיות שיוחזרו מטבלת מסד הנתונים, ניתן לכלול משפט Where של SQL. לדוגמה, התסריט הבא יחזיר רק את הרשומות בהן הערך בעמודת הסיסמה הוא apple:

```
<%
MySQL="SELECT * FROM Webusers WHERE password='apple' "
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
```

```
Set RS = Con.Execute( MySQL )
While Not RS.EOF
    Response.Write RS( "username" ) & " = " & RS( "password" ) & "<br>"
    RS.MoveNext
WEND
%>
```

לבסוף, אם ברצונך לאחזר מספר רשומות בלבד מטבלת מסד נתונים, תוכל להשתמש במילת המפתח TOP. התסריט הבא יאחזר עשר רשומות בלבד מהטבלה Webusers :

```
<%
MySQL="SELECT TOP 10 * FROM Webusers"
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Set RS = Con.Execute( MySQL )
While Not RS.EOF
    Response.Write RS( "username" ) & " = " & RS( "password" ) & "<br>"
    RS.MoveNext
WEND
%>
```

**הערה:**

מילת המפתח TOP אינה מוכרת על ידי גירסה 6.5 של SQL Server. ניתן להשתמש במילת מפתח זו רק בעבודה עם גירסה 7.0 של SQL Server, או בעבודה עם Microsoft Access.

# שיטות ומאפיינים מתקדמים של האובייקט Connection

בשני הסעיפים הבאים, תלמד על כמה מהתכונות המתקדמות של האובייקט Connection. תלמד כיצד להפעיל סדרה של משפטי SQL כתנועה (Transaction) יחידה. תלמד גם כיצד להשתמש באובייקט Connection לאחזור מידע אודות מסדי הנתונים והטבלאות הקיימות בשרת מסד הנתונים.

## שימוש ב-Transactions

תנועה (Transaction) היא רצף של פעולות שאמורות להתבצע בשלמותן. אם פעולה אחת נכשלת כל הפעולות נכשלות יחד איתה. כאשר אתה מפעיל סדרה של משפטי SQL במסגרת תנועה, מובטח לך שכל המשפטים יופעלו, או שאף אחד מהם לא יופעל. לדוגמה, נניח וברשותך תסריט להעברה של כל הרשומות מטבלה אחת לאחרת :

```
<%
Set Con = Server.CreateObject( "ADODB.Connection"  )
Con.Open "FILE NAME=C:\myDataLink.UDL"
Con.Execute "INSERT Webusers2 SELECT * FROM Webusers"
Con.Execute "Delete Webusers"
%>
```

משפט ה-SQL הראשון המופעל בתסריט מעתיק את כל הרשומות מהטבלה Webusers לטבלה Webusers2. לאחר מכן, נמחקות כל הרשומות בטבלה Webusers.

עכשיו, נניח וקרה דבר איום בעת הפעלת התסריט. למשל, דמיין לעצמך כי משפט ה-SQL הראשון הופעל, ומייד לאחר מכן נפגע שרת מסד הנתונים שלך מכוכב שביט. במצב כזה, משפט ה-SQL השני לא יופעל לעולם. נוצר מצב בו כל הרשומות הועתקו מטבלה אחת לשנייה, אך הרשומות בטבלה המקורית לא נמחקו. הטבלאות ייצאו מסנכרון.

כדי למנוע מקרים מסוג זה, ניתן להפעיל סדרה של משפטי SQL מתוך תנועה יחידה. לפניך עדכון של התסריט הקודם, כך שייעשה שימוש בתנועה:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection"  )
Con.Open "FILE NAME=C:\myDataLink.UDL"
Con.BeginTrans
Con.Execute "INSERT Webusers2 SELECT * FROM Webusers"
Con.Execute "Delete Webusers"
Con.RollBackTrans
%>
```

כשתפעיל תסריט זה, לא יבוצע כל שינוי במסד הנתונים. הרשומות לא יועתקו מטבלה אחת לשנייה, ורשומות הטבלה Webusers לא יימחקו. השיטה RollBackTrans מונעת מהמשפטים הכלולים בתנועה מלפעול.

**הערה:**

קיימות שתי דרכים נוספות לשימוש בתנועות בדפי ASP. ניתן ליצור תסריט ASP של תנועה על ידי שימוש בדפי ASP עם Microsoft Transaction Server. כמו כן, ניתן ליצור תנועות (transactions) מתוך שגרות מאוחסנות (Stored Procedures) של Microsoft SQL Server.

שגרות מאוחסנות הן פונקציות אשר מאוחסנות בשרת SQL ומכילות בתוכן משפטי SQL שמופעלים כאשר קוראים בשם הפונקציה דרך משפט SQL. (נושא זה יידון בהרחבה בהמשך הספר).

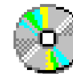
## עבודה עם Database Schemata

האובייקט Connection כולל שיטה שימושית מאד לאחזור מידע אודות מאפייני מסד הנתונים. על ידי שימוש בשיטה OpenSchema, ניתן לאחזר מידע אודות שמות מסדי הנתונים, הטבלאות הכלולות במסדי הנתונים, ומאפייני העמודות והאינדקס שלהם.

**הערה:**

אפשרות חלופית לשימוש בשיטה OpenSchema לאחזור מידע אודות מסד הנתונים היא שימוש ב-ADOX (הרחבות ADO לתמיכה ב-DLL ובאבטחה). למידע נוסף בעניין, עיין בתיעוד ADOX הכלול בחבילה הניתנת להורדה של MDAC.

השיטה OpenSchema מקבלת שני פרמטרים: סוג שאילתה (Query Type) וקריטריון (Criteria). פרמטר סוג השאילתה מציין את סוג אובייקט מסד הנתונים עליו ברצונך לקבל מידע. פרמטר הקריטריון מגביל את התוצאות המוחזרות מן השאילתה. השיטה OpenSchema מחזירה ערכת רשומות (RecordSet).



לדוגמה, כדי להחזיר רשימה של כל הטבלאות שנוצרו על ידי משתמשים (User-Created Tables), הנמצאות במסד הנתונים Pubs (מסד הנתונים לדוגמה הכלול ב-Microsoft SQL Server), ניתן להשתמש בתסריט הבא (הנמצא גם בתקליטור המצורף, בקובץ openSchema.asp):

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
queryType = adSchemaTables
criteria = Array( "Pubs", Empty, Empty, "TABLE" )
Set RS = Con.OpenSchema( queryType, criteria )
While NOT RS.Eof
  Response.Write RS( 2 ) & "<br>"
RS.MoveNext
Wend
%>
```

לתסריט זה מספר תכונות הדורשות הסבר. השורה הראשונה בתסריט כוללת קובץ בשם adovbs.inc. קובץ זה נכלל בתסריט על ידי שימוש בהפניית #INCLUDE של ASP. בכל מקרה של שימוש בקבועי ADO (ADO Constants), דוגמת הקבוע adSchemaTables בתסריט זה, יש לכלול קובץ זה.

**הערה:**

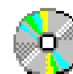
הקובץ adovbs.inc הותקן בשרת שלך בעת התקנת דפי ASP. קובץ זה מכיל רשימה ארוכה של קבועי ADO (ADO Constants) וערכיהם. כדי להפוך את הכללתו של קובץ זה בתסריטי ASP לנוחה יותר, רצוי להעתיקו לספריית השורש של שרת האינטרנט.

התסריט מקצה את המשתנה queryType לערך הקבוע adSchemaTables. כאשר השיטה OpenSchema נקראת עם סוג שאילתה זה, מוחזרת רשימה של טבלאות. רשימת טבלאות זו כוללת טבלאות שנוצרו על ידי משתמשים (User-Creaed Tables), טבלאות מערכת (System Tables), ותצוגות טבלה (Table Views).

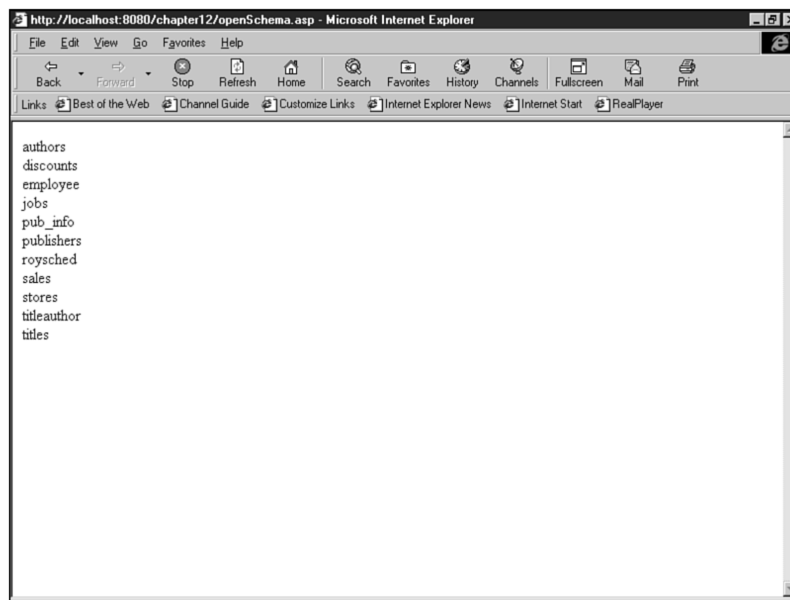
משתנה הקריטריון משמש להגבלת תוצאות השאילתה. ערכו של משתנה זה הוא מערך. המערך כולל ארבעה אלמנטים, המייצגים את בעל מסד הנתונים (Database Owner), שמו, וסוג הטבלה שיש להחזיר. בתסריט שלנו, מוחזרת רשימת הטבלאות ממסד הנתונים Pubs. הערך TABLE משמש להגבלת תוצאות השאילתה לטבלאות שנוצרו על ידי משתמשים בלבד.

לבסוף, לולאת WHILE..WEND משמשת להצגת רשימת הטבלאות (ראה תרשים 12.5). העמודה השנייה בערכת הרשומות (Recordset) מוצגת כדי להראות את שמות הטבלאות.

לשיטה OpenSchema סוגים רבים של קריטריונים וסוגי שאילתות.



כדי להדגים כמה מסוגי השאילתה הנוספים הניתנים לשימוש בשיטה OpenSchema, הבאנו דוגמה ליישום מסובך מעט יותר. קוד היישום מוצג בתדפיס 12.2. דף ASP זה נקרא DBSummary.asp, וניתן למצוא אותו גם בתקליטור המצורף.




**תרשים 12.5** שימוש בשיטה OpenSchema

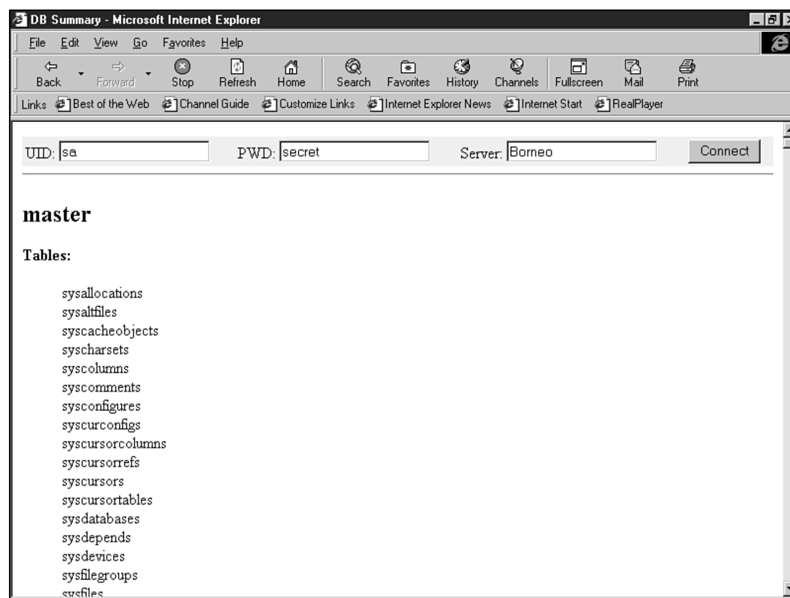
**תדפיס 12.2** DBSummary.asp

```
<!-- #INCLUDE VIRTUAL= "/adovbs.inc" -->
<%
On Error Resume Next
' Show All Tables In A Database
SUB showTables( byVal theDatabase, byRef theCon )
    Constraints = Array( theDatabase, Emtpy, Empty , TABLE )
    SET tableRS = Con.OpenSchema( adSchemaTables, Constraints )
    IF tableRS.Eof Then
        Response.Write "None"
    Else
    While NOT tableRS.Eof
        Response.Write tableRS( 2 ) & "<br>"
        tableRS.MoveNext
    Wend
    End IF
END SUB

' Show All Procedures In A Database
SUB showProcedures( byVal theDatabase, byRef theCon )
    Constraints = Array( theDatabase, Emtpy, Empty , Empty )
    SET procRS = Con.OpenSchema( adSchemaProcedures, Constraints )
    IF procRS.Eof Then
        Response.Write "None"
```

```
    Else
    While NOT procRS.Eof
        Response.Write procRS( 2 ) & " " &  "<br>"
        procRS.MoveNext
    Wend
    End IF
END SUB

UID = TRIM( Request( "UID" ) )
PWD = TRIM( Request( "PWD" ) )
DBServer = TRIM( Request( "DBServer" ) )
IF UID <> "" and DBServer <> "" THEN
    conString = "DRIVER={SQL Server}"
    conString = conString & ";SERVER=" & DBServer
    conString = conString & ";UID=" & UID
    conString = conString & ";PWD=" & PWD
    Set Con = Server.CreateObject( "ADODB.Connection" )
    Con.Open conString
    IF Err <> 0 THEN
        conString = ""
    End IF
END IF
%>
<HTML>
<HEAD><TITLE>DB Summary</TITLE></HEAD>
<BODY>
<FORM METHOD="post" ACTION="DBSummary.asp">
<TABLE BGCOLOR="#eeeeee" WIDTH="100%">
<TR>
    <TD>UID: <INPUT NAME="uid" SIZE=20 VALUE="<%=UID%>"></TD>
    <TD>PWD: <INPUT NAME="pwd" SIZE=20 VALUE="<%=PWD%>"></TD>
    <TD>Server: <INPUT NAME="DBServer" SIZE=20
        ↳ VALUE="<%=DBServer%>"></TD>
    <TD><input type="submit" value="Connect">
</TR>
</TABLE>
<HR>
<% IF conString = "" then %>
    No database specified or error while trying to connect.
<% else%>
<%
SET dbRS = Con.OpenSchema( adSchemaCatalogs )
While NOT dbRS.Eof
    DBName = dbRS( 0 )
    %>
    <H2><%=dbRS( 0 )%></H2>
    <B>Tables:</B>
    <BLOCKQUOTE><% showTables DBName, Con  %></BLOCKQUOTE>
```

```
    <B>Stored Procedures:</B>
    <BLOCKQUOTE><% showProcedures DBName, Con  %></BLOCKQUOTE>
<%
dbRS.MoveNext
WEND
%>
<% end if %>
</BODY>
</HTML>
```

DBSummary.asp מציג רשימה של כל מסדי הנתונים, הטבלאות, והשגרות המאוחסנות (Stored Procedures) הנמצאים בשרת מסד הנתונים. החיבור לשרת מתבצע על ידי אספקת שם כניסה של SQL (SQL Login), סיסמה, ושם מסד הנתונים (ראה תרשים 12.6).




**תרשים 12.6** הצגת סיכום אודות מסד נתונים

דף ASP זה יתחבר למסד נתונים ללא תלות במיקומו. ניתן להשתמש בתסריט גם ליצירת חיבור (connection) ואחזור מידע אודות מסד נתונים הנמצא על גבי האינטרנט. התסריט יוצר חיבור ODBC ללא DSN "על המקום", כדי להתחבר למסד נתונים.

# הבנת sessions
# ו-Connection Pooling

פתיחת חיבור (Connection) למסד נתונים גוזלת זמן ומאמץ ניכרים מדף ה-ASP. אם מאות אנשים ינסו לגשת בו זמנית לדף ASP הפותח חיבור למסד נתונים, הדבר עלול להעמיס במידה ניכרת על משאבי השרת שלך.

למרבה המזל, הוסיפה Microsoft הן ל-OLE DB והן ל-ODBC תכונה חדשה להקלה בעומס זה. OLE DB תומך ב-Session Pooling, ו-ODBC תומך ב-Connection Pooling.

כאשר החיבורים אגורים, נשארים חיבורי מסד הנתונים פתוחים, גם לאחר שדף ה-ASP שביקש את החיבור סיים את פעולתו. כאשר יתבקש חיבור חדש, ניתן יהיה למשכו מהמאגר. כך נמנע ההליך גוזל הזמן של פתיחת החיבור.

אין צורך לנקוט בפעולה מיוחדת כדי להשתמש באגירת חיבורים (Connection Pooling). התכונה מתאפשרת כברירת מחדל בהתקני ספקי OLE DB לשרת SQL וספק OLE DB ל-ODBC. המצב היחיד בו יהיה עליך לאפשר אגירת חיבורים באופן מפורש, הוא בעת שימוש ב-Internet Information Server 3.0 עם חיבור ODBC (כדי ללמוד כיצד לאפשר אגירת חיבורים במצב זה, עיין בתיעוד IIS 3.0).

**הערה:**

ניתן לנהל אגירת חיבורי ODBC מתוך ODBC Data Source Administrator, הנמצא בלוח הבקרה (Control Panel). לאחר פתיחת יישומון זה, בחר בכרטיסיה Connection Pooling.

כדי להפיק את המירב מאגירת החיבורים, עליך לזכור שלושה דברים.

דבר ראשון - לאחר פתיחת חיבור (connection) מתוך דף ASP, עליך לסגור אותו מוקדם ככל האפשר, כדי לשחררו בחזרה למאגר החיבורים (connections pooling). לא ניתן לעשות שימוש מחודש בחיבור שלא נסגר. השתמש בשיטה Close של האובייקט Connection כדי לסגור חיבור לאחר סיום השימוש בו.

דבר שני - ניתן לאגור רק חיבורים החולקים מחרוזת חיבור משותפת. לדוגמה, אם תפתח חיבור עם ערך UID חדש, לא יהיה ניתן לאגור אותו עם חיבורים המשתמשים בערך UID אחר.

כדי להבטיח כי דפי ASP שונים ישתמשו באותה מחרוזת חיבור לפתיחת חיבורים, יש להציב את ערך מחרוזת החיבור במשתנה Application. כדי לעשות זאת, שנה את השגרה Application_OnStart שבקובץ Global.asa כך שתכיל את המשפט :

```
Application( "conString" ) = "FILE NAME=C:\myDataLink.UDL"
```

ומעתה, בכל פעם שתפתח חיבור (connection) בדף ASP, פתח אותו עם משתנה היישום (Application Variable) המכיל את מחרוזת החיבור, כך :

```
Set Con  = "Server.CreateObject( "ADODB.Connection" )
Con.Open Application( "conString" )
```

דבר שלישי - יש להיות ערים לכך שאגירת חיבורים עלולה ליצור בעיות בעת שימוש במספר מאפיינים תלויי-חיבור של שרת SQL. לדוגמה, שגרות מאוחסנות וטבלאות זמניות הנוצרות בשרת SQL, נוצרות ביחס לחיבור מסוים. אובייקטים אלה אמורים להימחק באופן אוטומטי בעת סגירת החיבור. במקרה של שימוש באגירת חיבורים, ייתכן והאובייקטים לא יימחקו כראוי. דבר זה עלול להביא לידי כך שמסד הנתונים TempDB שלך יתמלא בערכי רפאים של שגרות מאוחסנות וטבלאות (ערכי רפאים הם ערכים שאינם בשימוש, אך ממשיכים לגזול זיכרון יקר).

בעיה זו אקוטית במיוחד בעת שימוש בחיבור ODBC, ולא בספק OLE DB טבעי. בעת יצירת מקור נתונים (Data Source) חדש באמצעות ה-ODBC Data Source Adminitrator, אל תבחר באפשרות ליצירת שגרות מאוחסנות זמנית (Temporary Stored Procedures) עבור משפטי SQL מוכנים (Prepared SQL Statements). אם תבחר באפשרות זו, ייתכן ויווצרו שגרות מאוחסנות זמנית שיישארו עמך לעד.

# סיכום

בפרק זה, למדת כיצד לפתוח חיבור (connection) למסד נתונים דוגמת Microsoft SQL Server או Microsoft Access. למדת כיצד לפתוח חיבורים (connections) הן באמצעות ספק OLE DB טבעי והן באמצעות ספק OLE DB להתקני ODBC. כמו כן, למדת כיצד להפעיל משפטי SQL באמצעות האובייקט Connection. לבסוף, למדת כיצד להשתמש במספר מאפיינים מתקדמים של האובייקט Connection.

## פרק 13

# עבודה עם Recordset

**בפרק זה:**

- **אחזור Recordset**
- **סוגי סמנים ונעילות באובייקט Recordset**
- **שיטות ומאפיינים מתקדמים באובייקט Recordset**

בפרק זה, תלמד לעבוד עם רשומות המאוחזרות ממסד הנתונים. תלמד כיצד להשתמש באובייקט Recordset של ADO כדי להציג רשימה של רשומות מתוך דף ASP. תלמד כיצד לעבוד עם Recordsets המשתמשות בסוגי סמן ונעילה שונים. לבסוף, תלמד אודות חלק מהשיטות והמאפיינים המתקדמים באובייקט Recordset.

**הערת העורך:**

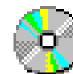
מסד נתונים מכיל רשומות. כל **רשומה** מכילה **שדות**. מכיון שאנו מדברים על מסדי נתונים טבלאיים, מתייחסים לאוסף של רשומות כאל **טבלה**, לרשומה בתור **שורה** ולשדה בתור **עמודה**.

# קבלת רשומות מבסיס נתונים באמצעות אובייקט Recordset

בפרק הקודם, למדת כיצד להשתמש באובייקט Connection של ADO להפעלת משפטי SQL. כשאתה מפעיל משפט SQL המחזיר רשומות מטבלת מסד נתונים, הרשומות מוחזרות מן המסד עם אובייקט ADO מסוג Recordset. ניתן להשתמש ב-Recordset כדי לייצג את הרשומות המוחזרות משאילתת SQL מתוך דף ASP.

אובייקט Recordset נועד להחזיק בתוכו את המידע שהתקבל ממסד הנתונים. אפשר לחשוב על אובייקט זה והמידע שבתוכו כמו טבלה המחזיקה בתוכה את הרשומות שהן תוצאת השאילתה שהופעלה על מסד הנתונים.

ניתן להשתמש באובייקט Recordset לייצוג של רשומה בודדת, או לייצוג של רבבות, ואפילו מיליוני רשומות. תהיה כמות הנתונים המוחזרים מן השאילתה גדולה או קטנה ככל שתהיה, הגישה לנתונים תתבצע באמצעות אובייקט זה.



לדוגמה, דף ה-ASP שבתדפיס 13.1 (הקוד נמצא גם בתקליטור המצורף, בקובץ showTable.asp) מאחזר את כל הרשומות מטבלת מסד הנתונים בשם Authors, ומציג את הרשומות, בזו אחר זו, לחלון הדפדפן (ראה תרשים 13.1).

Show Table - Microsoft Internet Explorer

| | | |
|---|---|---|
| Bennet | Abraham | 415 658-9932 |
| Blotchet-Halls | Reginald | 503 745-6402 |
| Carson | Cheryl | 415 548-7723 |
| DeFrance | Michel | 219 547-9982 |
| del Castillo | Innes | 615 996-8275 |
| Dull | Ann | 415 836-7128 |
| Green | Marjorie | 415 986-7020 |
| Greene | Morningstar | 615 297-2723 |
| Gringlesby | Burt | 707 938-6445 |
| Hunter | Sheryl | 415 836-7128 |
| Karsen | Livia | 415 534-9219 |
| Locksley | Charlene | 415 585-4620 |
| MacFeather | Stearns | 415 354-7128 |
| McBadden | Heather | 707 448-4982 |
| O'Leary | Michael | 408 286-2428 |
| Panteley | Sylvia | 301 946-8853 |
| Ringer | Albert | 801 826-0752 |
| Ringer | Anne | 801 826-0752 |

**תרשים 13.1** הצגת Recordset

**תדפיס 13.1** showTable.asp

```
<HTML>
<HEAD><TITLE>Show Table</TITLE>
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=Pubs"
Set RS = Con.Execute( "select * from Authors ORDER BY au_lname" )
%>
<TABLE BORDER=1>
<% While NOT RS.EOF %>
    <TR>
        <TD> <%=RS( "au_lname" )%> </TD>
        <TD> <%=RS( "au_fname" )%> </TD>
        <TD> <%=RS( "phone" )%> </TD>
    </TR>
<%
RS.MoveNext
Wend
RS.Close
%>
</TABLE>
</BODY>
</HTML>
```

דף ASP זה פותח חיבור (connection) למסד נתונים על ידי שימוש בקובץ קישור נתונים (Data Link) בשם myDataLink.UDL. לאחר פתיחת החיבור, מתבצעת שאילתת SQL מסוג SELECT, המאחזרת את כל הרשומות מהטבלה Authors, הנמצאת במסד הנתונים pubs (מסד הנתונים לדוגמה המגיע עם SQL Server). הרשומות מאוחזרות לתוך Recordset בשם RS, ומוצגות בזו אחר זו בלולאת WHILE...WEND.

שים לב כי לכל אורך התסריט לא נוצר מופע מפורש של האובייקט Recordset. Recordset מוחזר מהאובייקט Connection באופן אוטומטי, בכל ביצוע של שאילתת SQL.

### הערה:

קיימים מצבים בהם תצטרך להגדיר אובייקט Recordset באופן מפורש. דוגמה למצב כזה היא בעת עבודה עם Recordsets הניתנות לעדכון (Updateable Recordsets). ראה **סוגי סמנים ונעילות באובייקט Recordset**, בהמשך פרק זה.

לאחר אחזור רשומות הטבלה ל-Recordset, מוצגות העמודות au_lname, au_fname, ו-phone בכל רשומה, באמצעות לולאת WHILE...WEND. המאפיין EOF של האובייקט Recordset משמש כדי לזהות הגעה לרשומה האחרונה ב-Recordset. לאחר הצגת הרשומה האחרונה, נקבע ערך המאפיין EOF ל-TRUE, ופעולת הלולאה WHILE...WEND מסתיימת.

בתוך הלולאה WHILE...WEND, נעשה שימוש בשיטה MoveNext של האובייקט Recordset, למעבר לרשומה הבאה. בפעם הראשונה בה נפתח Recordset, מייצג האובייקט Recordset את הרשומה הראשונה שהוחזרה מהשאילתה. לאחר קריאה לשיטה MoveNext, מייצג האובייקט Recordset רשומה חדשה.

לבסוף, לאחר הצגת כל הרשומות, נסגר Recordset על ידי קריאה לשיטה Close של האובייקט Recordset. Recordset נסגר באופן אוטומטי כשדף ASP מסיים את פעולתו, אך סגירת ה-Recordset בשלב מוקדם משחררת משאבי מערכת בשרת.

**הערה:**

יש להזהר בעת שימוש ב-Recordset לאחזור ערכי טקסט. כדי למנוע החזרה של רשומות ריקות, יש לציין את עמודת הטקסט כעמודה האחרונה במשפט ה-SELECT, או לפתוח סמן אחר במקום סמן ברירת המחדל קדימה בלבד (Default Forward-Only Cursor). למידע נוסף אודות סמנים, ראה **סוגי סמנים ונעילות באובייקט Recordset**, בהמשך פרק זה.

# שדות אובייקט Recordset

כל Recordset מכיל אוסף (Collection) בשם Fields. האוסף Fields מכיל אובייקטי Field. השדות באוסף Field מייצגים את העמודות שהוחזרו משאילתת מסד הנתונים. קיימות מספר דרכים לשימוש באוסף Fields להצגת ערך עמודה.

הדרך הראשונה והנפוצה ביותר היא הצגת ערך של שדה יחיד על ידי שימוש בשם השדה. לדוגמה, התסריט הבא מציג עמודה בשם phone על ידי שימוש בשמה :

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDatalink.UDL;DATABASE=pubs"
Set RS=Con.Execute( "SELECT phone FROM Authors WHERE au_lname='Green' ")
Response.Write RS( "phone" )
%>
```

תסריט זה מאחזר שדה ברשומה מטבלת מסד הנתונים pubs, ומציג את ערך השדה. התסריט מאחזר את מספר הטלפון של הסופר ששם משפחתו Green, ומציג אותו.

הערך מוצג בשורה האחרונה בתסריט:

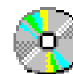
```
Response.Write RS( "phone" )
```

משפט זה מציג את ערך השדה (Field) על ידי העברת שם השדה לאובייקט Recordset. עם זאת, ניתן להציג את אותו הערך גם על ידי שימוש בכל אחד מהמשפטים הבאים:

```
Response.Write ( "phone" )
Response.Write RS.Fields( "phone" )
Response.Write RS.Fields( "phone" ).Value
Response.Write RS( 0 )
Response.Write RS.Fields( 0 )
Response.Write RS.Fields( 0 ).Value
```

האוסף Fields הוא אוסף ברירת המחדל של האובייקט Recordset. משמעות הדבר היא שניתן לפנות לשדה מסוים על ידי פנייה מפורשת לאוסף Fields, או על ידי פנייה לאובייקט Recordset המכיל את האוסף Fields. מומלץ להשתמש בקו הפעולה השני (פנייה לאובייקט), מפני שהוא חוסך הקלדה.

שלושת המשפטים האחרונים מציגים את ערך השדה על ידי פנייה למקומו באוסף Fields. במקרה זה, היות ושאילתת ה-SQL אחזרה ערך אחד בלבד, מוצג ערך העמודה phone.



מדוע לאחזר את ערך השדה על פי מיקומו, ולא על פי שמו? שימוש במיקום השדה יעיל כאשר אינך יודע את שמות העמודות בטבלת מסד נתונים. לדוגמה, דף ASP שבתדפיס 13.2 (ניתן למצאו גם בתקליטור המצורף, בקובץ anytable.asp) עושה שימוש בשדה בטופס HTML כדי לקבל שם של טבלה. לאחר קבלת שם הטבלה, מוצגות כל השורות והעמודות שבטבלה המבוקשת (ראה תרשים 13.2).

Table Name: Titles Display

| title_id | title | type | pub_id | price | advance | royalty | ytd_sales | notes | pubdate |
|---|---|---|---|---|---|---|---|---|---|
| BU1032 | The Busy Executive's Database Guide | business | 1389 | 19.99 | 5000 | 10 | 4095 | An overview of available database systems with emphasis on common business applications. Illustrated. | 6/12/91 |
| BU1111 | Cooking with Computers: Surreptitious Balance Sheets | business | 1389 | 11.95 | 5000 | 10 | 3876 | Helpful hints on how to use your electronic resources to the best advantage. | 6/9/91 |
| BU2075 | You Can Combat Computer Stress! | business | 0736 | 2.99 | 10125 | 24 | 18722 | The latest medical and psychological techniques for living with the electronic office. Easy-to-understand explanations. | 6/30/91 |
| | Straight Talk | | | | | | | Annotated analysis of what computers can | |

**תרשים 13.2** הצגה אוטומטית של כל Recordset

**תדפיס 13.2** anyTable.asp

```
<HTML>
<HEAD><TITLE>Any Table</TITLE></HEAD>
<BODY>
<FORM METHOD="post" action="AnyTable.asp">
Table Name: <INPUT NAME="tablename" SIZE=20>
<INPUT TYPE="submit"VALUE="Display">
</FORM>
<HR>
<%
tablename = TRIM( Request( "tablename" ) )
if tablename <> "" then
    Set Con = Server.CreateObject( "ADODB.Connection" )
    Con.Open "FILE NAME=c:\myDataLink.UDL"
    Set RS = Con.Execute( "SELECT * FROM " & tablename )
    %>
    <TABLE BORDER=1>
        <TR>
        <% for i = 0 to RS.Fields.Count - 1 %>
        <TH><%=RS( i ).name%></TH>
        <% next %>
        </TR>
    <% while not RS.EOF %>
        <TR>
        <% for i = 0 to RS.Fields.Count - 1 %>
        <TD><%=RS( i ).value %> </TD>
```

```
        <% next %>
        </TR>
    <% RS.MoveNext %>
    <% wend %>
    </TABLE>
<%
end if
%>
</BODY>
</HTML>
```

לאחר קבלת שם הטבלה, מציג התסריט את שמות העמודות באמצעות לולאת FOR...NEXT. שמות העמודות מוחזרים על ידי שימוש במאפיין Name של האובייקט Field.

לאחר מכן, מוצגים ערכי כל העמודות שבכל שורה, על ידי שימוש בלולאת FOR...NEXT נוספת. לולאה זו עוברת על כל העמודות בזו אחר זו, ומציגה את ערכן. על ידי פנייה ל-Recordset Fields לפי מיקומם, מוצג כל המידע שבטבלה, מבלי לדעת דבר על מבנה.

לאובייקט Field מספר מאפיינים שימושיים נוספים, הניתנים לשימוש בתסריטי ASP לאחזור מידע אודות העמודות שהוחזרו משאילתת מסד נתונים. מאפיינים אלה מפורטים בטבלה 13.1.

**טבלה 13.1** מאפיינים באובייקט Field

| מאפיין | שימוש |
|---|---|
| ActualSize | משמש להחזרת אורך המידע המאוחסן בעמודה, במיקרים של שימוש בעמודות בהן סוג הנתונים הוא בעל אורך משתנה. למשל, עמודות VARCHAR. |
| Attributes | כולל מספר מאפיינים שונים של עמודות. לדוגמה, ניתן להשתמש במאפיין זה כדי לקבוע האם העמודה מכילה ערכי NULL, או האם העמודה מכילה סוג נתונים בעל אורך קבוע. |
| DefinedSize | מספר הבתים (Bytes), או התווים, המקסימלי שיכולה העמודה להכיל. לדוגמה, ייתכן וערך DefinedSize (גודל מוצהר) של עמודה מסוג VARCHAR יהיה 500, וערך ActualSize (גודל מעשי) יהיה 2. |
| Name | שם העמודה. |
| NumericScale | עבור עמודות עם נתונים מספריים (Numeric) או עשרוניים (Decimal), מחזיר המאפיין את קנה המידה של העמודה (מספר הספרות המקסימלי שיכול להופיע מימין לנקודה העשרונית). |

| מאפיין | שימוש |
|---|---|
| OriginalValue | בשדה הניתן לעדכון, מכיל המאפיין את ערך שדה ב-Recordset המקורי, לפני ששונה. בהמשך הפרק, בסעיף **עדכון רשומות באמצעות Navtive ADO Methods**, נידון בהרחבה בנושא עדכון של Recordsets. |
| Precision | עבור עמודות עם נתונים מספריים (Numeric) או עשרוניים (Decimal), מחזיר מאפיין זה את דיוק העמודה (מספר הספרות המקסימלי שיכול להופיע מימין ומשמאל לנקודה העשרונית). |
| Type | סוג הנתונים של העמודה - לדוגמה, האם העמודה היא עמודת date (תאריך) או עמודת VARCHAR. |
| UnderlyingValue | מאפיין זה מכיל את ערכה הנוכחי של העמודה, המופיע בטבלת מסד הנתונים. ערך מאפיין זה עלול להיות שונה מערכו של המאפיין OriginalValue, כיון שייתכן והמידע בעמודת מסד הנתונים עודכן לאחר אחזור Recordset. |
| Value | ערך העמודה. אין צורך להשתמש במאפיין זה באופן מפורש, היות והוא מאפיין ברירת המחדל באובייקט Field. |

# סוגי סמנים ונעילות באובייקט Recordset

בכל פתיחה של Recordset, ה-Recordset נפתח - באופן מפורש או בלתי מפורש - עם סוג מסוים של סמן ושל נעילה (Particular Cursor and Locking Type). הבנת סוגי הסמנים (Cursors) והנעילות (Locking) חשובה, משתי סיבות:

- ראשית, שימוש בסוג שונה של סמן או של נעילה עלול להשפיע באופן דרמטי על ביצועי תסריט ASP שלך. אם תבחר באפשרות הלא נכונה, עלול דף ASP להתעכב במשך זמן רב לפני שיאחזר ויציג את רשומות מסד הנתונים.
- שנית, מאפיינים מסוימים של האובייקט Recordset זמינים רק בעת שימוש בסוג הנעילה והסמן הנכונים. לדוגמה, עליך לפתוח Recordset עם סוגי הסמן והנעילה הנכונים, אם ברצונך לאחזר את מספר הרשומות שב-Recordset, או ליצור Recordset המכילה רשומות הניתנות לעדכון.

בחירת סוגי הסמן והנעילה הנכונים עבור Recordset מצריכה איזון בין שני שיקולים: הצורך בביצועים חזקים, מול הצורך במאפיינים ספציפיים של ה-Recordset. לאחר קריאת הסעיפים הבאים, יהיה לך מושג טוב יותר לגבי ביצוע איזון זה.

# הבנת סמני ADO

כשאתה מפעיל שאילתת SQL המחזירה Recordset של רשומות, מועברות הרשומות ליישום ה-ADO דרך סמן (Cursor). קיימים ארבעה סוגים שונים של סמנים, הקובעים את אופן ההעברה של תוצאות השאילתה:

- סמן קדימה-בלבד (Forward-Only Cursor) - זהו הסמן ברירת המחדל, והוא מספק את הביצועים המהירים ביותר. כאשר Recordset נפתח עם סמן מסוג זה, התוצאות מאוחזרות באופן סדרתי, מהראשונה לאחרונה. סמני קדימה-בלבד (Forward-Only Cursors) אינם תומכים בגלילה (Scrolling) לאחור, ומאפשרים מעבר אחד בלבד על ערכת התוצאות.
- סמן סטטי (Static Cursor) - סמן זה משקף את מצב הנתונים בטבלה כאשר הוא נפתח לראשונה. אין לסמן זה אפשרות לזהות אם רשומות הטבלה שונו או נמחקו, או אם נוספו רשומות חדשות. אולם, בניגוד לסמן קדימה-בלבד, סמן זה מאפשר גלילה קדימה ואחורה בערכת התוצאות.
- סמן מונע ערכת מפתחות (Keyset Driven Cursor) - סמן זה מסוגל לזהות שינויים מסוימים ברשומות טבלת הנתונים, אך לא כל שינוי. בייחוד, מסוגל סמן זה לזהות במדויק נתונים שעודכנו. עם זאת, אין באפשרות הסמן לזהות האם משתמשים אחרים הוסיפו או מחקו רשומות (רשומות שהוסרו יוצרות חורים ב-Recordset). גם סמן זה תומך בגלילה.
- סמן דינמי (Dynamic Cursor) - זהו "הסמן". ביכולתו לזהות כל שינוי הנעשה בטבלה על ידי משתמשים אחרים, כל עוד הוא פתוח. כמו כן, סמן זה תומך בגלילה באופן מלא.

כאשר אתה פותח Recordset מבלי לציין את סוג הסמן, כפי שנעשה בדוגמאות הקודמות בפרק זה, נפתח Recordset עם סמן קדימה-בלבד. כאשר נעשה שימוש בסמן ברירת מחדל זה בצירוף עם שרת SQL, נפתח הסמן במצב Firehose (זרנוק). מצב זה נקרא כך, כיון שהחיבור (connection) נשאר פתוח, ותוצאות השאילתה מועברות ליישום במהירות האפשרית.

כדי להשתמש ב-Dynamic Cursor, יש ליצור מופע של האובייקט Recordset באופן מפורש, ולקבוע את סוג הסמן של האובייקט לפני הפעלת השאילתה. לפניך דוגמה לפתיחה של סמן דינמי:

```
<! -- #INCLUDE VIRTUAL="/adovbs.inc" -- >
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=pubs"
```

```
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.CursorType = adOpenDynamic
RS.Open "SELECT * FROM Authors", Con
%>
```

התסריט יוצר באופן מפורש מופע של האובייקט Recordset. הדבר הכרחי כדי לאפשר קביעה של ערך המאפיין CursorType. לאחר קביעת CursorType (סוג הסמן) לסמן דינמי, נפתח Recordset על ידי העברה של מחרוזת SQL, ושל שם אובייקט ה-Connection.

שים לב כי התסריט משתמש בהפניית INCLUDE של ASP, וכולל את הקובץ adovbs.inc. הדבר חיוני, היות ובתסריט נעשה שימוש בקבוע ADO (ADO Constant) בשם adOpenDynamic. לפניך רשימה של קבועי ADO עבור כל סוגי הסמנים:

| | |
|---|---|
| adOpenForwardOnly | קדימה בלבד |
| adOpenStatic | סטטי |
| adOpenKeyset | מונע ערכת מפתחות |
| adOpenDynamic | דינמי |

כעת, לאחר שהבנת את סוגי הסמנים השונים, כיצד תחליט באיזה סמן להשתמש בפתיחת Recordset? באופן כללי, כדאי להשתמש בסמן ברירת המחדל, שכן הוא המהיר ביותר. יש לשקול שימוש בסוגי סמן אחרים רק בעת עבודה עם כמה מהמאפיינים המתקדמים באובייקט Recordset של ADO, הנידונים בהמשך פרק זה.

## סמנים של צד-שרת ושל צד-לקוח

בעבודה עם סמני ADO קיימת אפשרות בחירה נוספת. ניתן לבחור האם לפתוח סמן צד-שרת (Server-Side Cursor), או סמן צד-לקוח (Client-Side Cursor). בשימוש בסמן צד-לקוח, הסמן נפתח ומנוהל במחשב המארח את דף ASP. בשימוש בסמן צד-שרת, הסמן נפתח ומנוהל במחשב המארח את מסד הנתונים.

כברירת מחדל, בזמן יצירת Recordset נעשה שימוש בסמן צד-שרת. כדי לציין שברצונך לעבוד עם סמן צד-לקוח, יש להשתמש במאפיין CursorLocation, ראה להלן:

```
<!--#INCLUDE VIRTUAL="/adovbs.inc"-->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=pubs"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.CursorLocation = adUseClient
RS.CursorType = adOpenDynamic
RS.Open "SELECT * FROM Authors", Con
%>
```

בדוגמה זו, מוקצה הקבוע adUseClient למאפיין CursorLocation, כדי לציין שימוש בסמן צד-לקוח. Recordset נפתח עם סמן דינמי של צד-לקוח (Dynamic Client-Side Cursor).

היתרון בשימוש בסמן צד-לקוח, הוא בהעברת עומס העבודה משרת מסד הנתונים. כל תוצאות השאילתה נשלחות לשרת שבתוכו נמצאים דפי ה-ASP.

שימוש בסמן צד-לקוח עלול גם להגביר את ביצועי יישום ה-ASP. היות והסמן והיישום נמצאים באותו המחשב, ניתן לדפדף ברשומות הסמן במהירות רבה.

יש להשתמש בסמני צד-לקוח במקרים בהם אינך רוצה להכביד יתר על המידה על שרת מסד הנתונים. מצד שני, במקרים בהם אינך רוצה להעמיס את שרת האינטרנט, או ליצור עומס על הרשת, יש להשתמש בסמן ברירת המחדל, הנמצא בצד השרת.

**הערה:**

בעת שימוש בסמן Keyset Driven Cursor בצד-לקוח, לא נשלחים הנתונים ללקוח. סמן Keyset Driven Cursor מייצג ערכת תוצאות על ידי ערכת מפתחות, לזיהוי ייחודי של כל רשומה. כאשר מתבקש סמן רשומה מסוים, נעשה שימוש במפתח לאחזור הרשומה המתאימה מטבלת מסד הנתונים.

# הבנת סוגי הנעילה של ADO

אתר אינטרנט המונע על ידי מסד נתונים, עלול להידרש לתמוך במאות משתמשים הניגשים לאותן הטבלאות באותו הזמן. במצב בו יותר ממשתמש אחד יקרא או יעדכן מידע מטבלת מסד נתונים בו זמנית, יתרחשו התנגשויות.

לדוגמה, נניח ואתר האינטרנט שלך מכיל טבלה המשמשת לאגירת מספר הביקורים בדף הבית של האתר. כאשר **תמי** - המבקרת ה-1,888 באתר - ניגשת לדף הבית, מאוחזר מהטבלה הערך 1,887. באותו הזמן, ניגש המבקר **אורן** לדף הבית גם הוא, וגם עבורו מאוחזר מהטבלה הערך 1,887. אם הערך יעודכן עבור שני המשתמשים בו זמנית, ייאגר ערך שגוי בטבלת מסד הנתונים.

כדי לשלוט באופן בו יטופלו התנגשויות אלו, ניתן להשתמש במאפיין LockType. הערכים האפשריים למאפיין זה מפורטים בטבלה 13.2 :

**טבלה 13.2** הערכים האפשריים למאפיין LockType

| ערך | הגדרה |
|---|---|
| AdLockReadOnly | ברירת המחדל של סוגי הנעילה. סוג הנעילה לקריאה בלבד (Read Only) משמש כדי לאפשר למספר משתמשים לקרוא את אותם הנתונים בו זמנית. עם זאת, בפתיחה של Recordset עם סוג נעילה זה, לא ניתן לשנות את הנתונים. |
| AdLockPessimistic | בפתיחה של Recordset עם נעילה פסימית (Pessimistic), המשתמשים האחרים מנועים מהגישה לנתונים, מרגע התחלת עריכת הרשימה. |
| AdLockOptimistic | בפתיחה של Recordset עם נעילה אופטימית (Optimistic), יכולים המשתמשים האחרים לגשת לנתונים, עד לרגע ביצוע העדכון. |
| AdLockBatchOptimistic | סוג נעילה זה משמש לביצוע עדכוני אצווה (Batch Updates). בסעיף **עדכון אצווה של רשומות** יוסבר סוג נעילה זה ביתר פירוט. |

אם דף ה-ASP שלך משמש רק להצגת נתונים מטבלת מסד הנתונים, אין צורך לשנות את ברירת המחדל לגבי סוגי הנעילה. אולם, אם בכוונתך להשתמש ב-Native ADO Methods כדי לעדכן נתונים, יהיה עליך לבחור בסוג נעילה אחר. באופן כללי, רצוי לבחור בנעילה האופטימית, היות והיא גורמת לכמות המועטה ביותר של התנגשויות עם משתמשים אחרים, והכי פחות סביר שתגרום לקריסת (או "תקיעת") מסד הנתונים.

### הערה:

כאשר משנים מידע במסד הנתונים על ידי הפעלת משפטי SQL, אין צורך לדאוג לסוג הנעילה. המאפיין LockType רלוונטי רק במקרים של עדכון מסד הנתונים תוך שימוש בשיטות מקומיות של ADO, דוגמת AddNew.

להלן דוגמה הקובעת את סוג הנעילה לנעילה אופטימית:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.LockType = adLockOptimistic
%>
```

# שיטות ומאפיינים מתקדמים באובייקט Recordset

בסעיפים הבאים, תלמד את אופן השימוש במספר מהתכונות המתקדמות של האובייקט Recordset. תלמד כיצד לבצע משימות כגון אחזור מספר הרשומות שב-Recordset, דפדוף וגלילה (Paging and Scrolling) בערכת נתונים, ויצירת Persistent Recordsets.

## אחזור מספר הרשומות

לאחר פתיחת Recordset, ניתן לגלות את כמות הרשומות שבה על ידי שימוש במאפיין RecordCount. מאפיין זה אינו נתמך בסמנים מסוג קדימה-בלבד (Forward Only) או בסמנים דינמיים (Dynamic Cursors). בעת שימוש בסמנים אלה, יהיה ערכו של מאפיין זה שווה ל- 1-.

לפניך דוגמה העושה שימוש במאפיין RecordCount תוך שימוש בסמן סטאטי (Static Cursor):

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL; DATABASE=pubs"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS,CursorType = adOpenStatic
RS.Open "SELECT * FROM Authors", Con
Response.Write "This Recordset contains:"
Response.Write RS.RecordCount
Response.Write " records"
%>
```

רצוי להימנע ככל האפשר משימוש במאפיין RecordCount, כיון שהוא דורש Dynamic Cursor. אם כל רצונך לבדוק האם ה-Recordset ריק, עדיף לעשות שימוש במאפיין EOF, כך:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL; DATABASE=pubs"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS,CursorType = adOpenStatic
RS.Open "SELECT * FROM Authors", Con
if RS.EOF
    Response.Write "No records retrieved"
else
Response.Write "Records retrieved!"
end if
%>
```

## גלילה באובייקט Recordset

האובייקט Recordset של ADO כולל מספר שיטות ומאפיינים לתנועה קדימה ואחורה ברשומות ה-Recordset. לדוגמה, התסריט הבא פותח Recordset, מדלג לרשומה האחרונה, ומציג את כל הרשומות שב-Recordset תוך תנועה לאחור - מהרשומה האחרונה לראשונה:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL; DATABASE=pubs"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS,CursorType = adOpenStatic
RS.Open "SELECT * FROM Authors ORDER BY au_lname", Con
ES.MoveLast
While not RS.BOF
Response.Write RS( "au_lname"  ) & "<br>"
RS.MovePrevious
Wend
%>
```

תסריט זה משתמש בשיטה MoveLast כדי לדלג לרשומה האחרונה, בשיטה MovePrevious למעבר לרשומה הקודמת, ובמאפיין BOF לגילוי ההגעה לרשומה הראשונה.

בעת גלילה ב-Recordset, ניתן להשתמש במאפיינים המוסברים בטבלה 13.3:

**טבלה 13.3** מאפיינים המשמשים ב-Recordset

| מאפיין | שימוש |
|---|---|
| AbsolutePosition | קביעה או החזרה של מיקומה הסידורי הנוכחי של הרשומה הנוכחית ב-Recordset. |
| Bookmark | קביעה או החזרה של סימניה לזיהוי ייחודי של הרשומה הנוכחית ב-Recordset. |
| BOF | מאפיין זה מקבל את הערך TRUE כאשר הרשומה הנוכחית ממוקמת לפני הרשומה הראשונה. |
| EOF | מאפיין זה מקבל את הערך TRUE כאשר הרשומה הנוכחית ממוקמת אחרי הרשומה האחרונה. |

המאפיין AbsolutePosition מאפשר מעבר ישיר לכל רשומה על פי מיקומה. לדוגמה, התסריט הבא מאחזר ומחזיר את הרשומה המופיעה באמצע Recordset :

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL; DATABASE=pubs"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS,CursorType = adOpenStatic
RS.Open "SELECT * FROM Authors ORDER BY au_lname", Con
RS.AbsolutePath = RS.RecordCount / 2
Response.Write( "au_lname" )
%>
```

המאפיין Bookmark משמש להחזרת מזהה ייחודי (Unique Identifier) לרשומה, כך שניתן יהיה לחזור אליה מאוחר יותר. לדוגמה, התסריט הבא יוצר סימניה לרשומה, ולאחר מכן קובע את סמן ה-Recordset כך שיצביע לסימניה :

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL; DATABASE=pubs"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS,CursorType = adOpenStatic
RS.Open "SELECT * FROM Authors ORDER BY au_lname", Con
While not RS.EOF
    if RS( "au_lname" ) = "Locksley" then
        ' Set the Bookmark
        myBookmark = RS.Bookmark
    end if
RS.MoveNext
Wend
' Return to the Bookmark
RS.Bookmark = myBookmark
Response.Write RS( "au_lname"  )
%>
```

בדוגמה זו, נוצרת סימניה בעת מציאת רשומה המכילה את הערך Locksley. לאחר ההגעה לסוף ה-Recordset, סמן ה-Recordset נקבע לרשומה זו, על ידי שימוש בסימניה.

לגלילה ב-Recordset, ניתן להשתמש בשיטות שבטבלה 13.4.

**טבלה 13.4** שיטות לשימוש ב-Recordset

| שיטה | שימוש |
|---|---|
| Move | מעבר למספר מסוים של רשומות. מספרים שליליים מובילים אחורה, וחיוביים קדימה. ניתן לנוע ביחס לרשומה הנוכחית, או ביחס לסימניה. |
| MoveFirst | מעבר לרשומה הראשונה ב-Recordset (קריאה לשיטה זו בעת שימוש בסמן מסוג קדימה-בלבד עלולה לגרום להפעלה מחודשת של שאילתת SQL). |
| MoveLast | מעבר לרשומה האחרונה ב-Recordset. |
| MoveNext | מעבר לרשומה הבאה ב-Recordset. |
| MovePrevious | מעבר לרשומה הקודמת ב-Recordset. |

שים לב כי שיטות ומאפיינים רבים המשמשים לגלילה ב-Recordset לא עובדים ביעילות, או לא עובדים כלל, בעת שימוש בסמן מסוג קדימה-בלבד (Forward Only). לדוגמה, השיטה MovePrevious אינה תואמת סמני קדימה-בלבד, ותגרום לשגיאה בעת ניסיון להשתמש בה בשילוב עם סמן מסוג זה.

**הערה:**

ניתן להשתמש בשיטה MovePrevious עם Recordset קדימה-בלבד, על ידי קביעת גודל המטמון (Cache Size) של ה-Recordset לערך הגדול מ-1. אופן פעולה זה אינו מומלץ, כיוון שהוא צורך זיכרון רב.

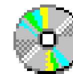
רצוי לצמצם ככל אפשר את השימוש בשיטות גלילה הדורשות שימוש ב-Dynamic Cursor. בדרך כלל, ניתן להשיג תוצאות זהות על ידי שימוש בשאילתות SQL טובות יותר. לדוגמה, כדי להציג את הרשומות שב-Recordset בסדר הפוך, ניתן להשתמש במשפט ה-SQL (SQL Clause) ORDER BY, במקום בשיטות MoveLast ו-MovePrevious של ADO.

## דפדוף ב-Recordset

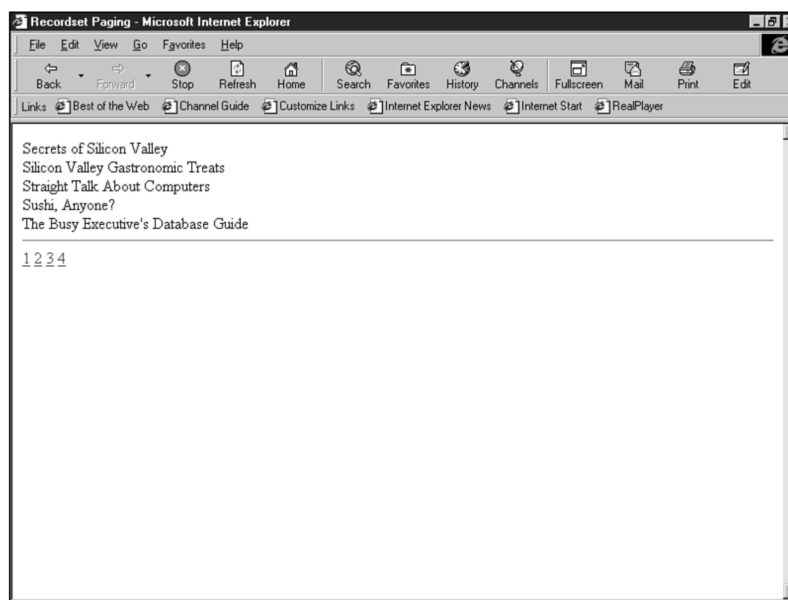
קיימים מצבים רבים בהם יהיה עליך לספק למשתמש אמצעים לדפדוף ברשומות שב-Recordset. לדוגמה, נניח כי עליך ליצור דף ASP המציג רשימת מוצרים למכירה. במקרה שאתה מוכר מאות מוצרים, ייתכן ותרצה לחלק את רשימת המוצרים על פני מספר דפים.

האובייקט Recordset מכיל מספר מאפיינים מיוחדים לדפדוף ב-Recordset. על ידי שימוש במאפיינים אלה לחלוקת הרשומות שב-Recordset לדפים שונים, תוכל להציג את נתוני ה-Recordset בחלקים. להלן רשימת מאפיינים אלה:

- AbsolutePage - מאפיין זה קובע או מחזיר את דף הרשומות הנוכחי.
- PageCount - מאפיין זה מחזיר את מספר הדפים שב-Recordset.
- PageSize - מאפיין זה קובע או מחזיר את מספר הרשומות הכלולות בדף יחיד (ברירת המחדל היא 10 רשומות בדף).



כדי לחלק Recordset לדפים, יש לקבוע את מספר הרשומות בדף על ידי שימוש במאפיין PageSize. לאחר מכן, לאחר פתיחת Recordset, יש להשתמש במאפיין AbsolutePage למעבר לעמוד ספציפי. תדפיס 13.3 (הנמצא בתקליטור המצורף, בקובץ Paging.asp) מציג דוגמה להצגת הרשומות שבטבלה בשם Titles, תוך שימוש במספר דפים (ראה תרשים 13.3).




**תרשים 13.3** Recordset המחולק על פני מספר דפים

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<HTML>
<HEAD><TITLE>Recordset Paging</TITLE></HEAD>
<BODY>
<%
DIM currentPage, rowCount, i
currentPage = TRIM( Request( "currentPage" ) )
if currentPage = "" then currentPage = 1
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=pubs"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.CursorType = adOpenStatic
RS.PageSize = 5
RS.Open "SELECT title FROM Titles ORDER BY title", Con
RS.AbsolutePage = cINT( currentPage )
rowCount = 0
While not RS.EOF and rowCount < RS.PageSize
    Response.Write RS( "title" ) & "<BR>"
    rowCount = rowCount + 1
    RS.MoveNext
Wend
Response.Write "<BR>"
for i = 1 to RS.PageCount
    %>
    <A HREF="Paging.asp?currentPage=<%=i%>"><%=i%></A>
    <%
next
%>
</BODY>
</HTML>
```

**הערה:**

הקובץ adovbs.inc הוא קובץ טקסט המכיל רשימה ארוכה של קבועים (Constants) המשמשים את ADO. הקובץ הותקן על השרת שלך בעת התקנת Internet Information Server. ניתן למצוא קובץ זה בתיקיה InetPub\iissamples\ISSamples שבשרת האינטרנט.

יש לשים לב לא לשכוח לכלול קובץ זה מכיון שכל הקבועים אשר מגדירים את הפעילויות החיוניות ב-Recordset נמצאים בו. אם נשכח לכלול קובץ זה, פעולות כמו מספר דפים, כמו בדוגמה הנ"ל, יכשלו!

מספר התייחסויות לגבי תדפיס 13.3 : קודם כל, שים לב לשימוש במשתנה rowCount להגבלת מספר הרשומות המוצגות בעמוד מסוים. לולא השימוש במשתנה זה, כל דף היה מכיל, בנוסף לרשומות המיועדות לו, גם את רשומות הדפים הבאים.

כמו כן, שים לב לפונקציה ()cINT של VBScript, המשמשת להמרת הערך המועבר במחרוזת URL למספר שלם (Integer). ללא השימוש בפונקציה זו ייתכן וערך המשתנה לא יפורש נכונה כמספר שלם, דבר העלול להוליד שגיאות משונות.

לבסוף, שים לב כי ה-Recordset לא נפתח עם סמן קדימה-בלבד (Forward Only). כדי להשתמש במאפייני הדפדוף, יש להשתמש ב-Dynamic Cursor. למרבה הצער, פירוש הדבר הוא שמאפייני הדפדוף אינם יעילים. בעת דפדוף ב-Recordset המכיל אלפי רשומות, אחזור ותצוגה של דף רשומות עלול להמשך זמן רב.

# הגבלת מספר הרשומות המוחזרות ב-Recordset

לעיתים, ייתכן ולא תרצה לאחזר את כל הרשומות משאילתת SQL. למשל, ייתכן ותרצה לאחזר את 15 הרשומות הראשונות שבטבלה בלבד, או רק 10 אחוז מהרשומות שבראש הטבלה. ניתן להגביל את מספר הרשומות המוחזרות משאילתת מסד נתונים בשתי דרכים: על ידי שימוש במאפיין MaxRecords של ADO, או על ידי שימוש במילת המפתח (Keyword) TOP של שרת SQL. התסריט הבא משתמש במאפיין MaxRecords לאחזור 15 סופרים בלבד מהטבלה Authors:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL; DATABASE=pubs"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.MaxRecords = 15
RS.Open "SELECT * FROM Authors ORDER BY au_lname", Con
While not RS.EOF
    Response.Write( "au_lname" ) & "<BR>"
    RS.MoveNext
Wend
%>
```

שים לב כי המאפיין MaxRecords אינו דורש שימוש בסוג סמן Rich. ניתן להשתמש במאפיין זה גם בעבודה עם Recordsets שנפתחו עם סמן קדימה-בלבד. כמו כן, שים לב לשימוש במשפט ORDER BY בשאילתת SQL. ללא שימוש במשפט זה, יוחזר אוסף אקראי של 15 רשומות. תסריט זה מאחזר את 15 הסופרים הראשונים, לפי סדר שמות משפחתם.

אם אתה משתמש בגירסה 7.0 של שרת SQL או ב-Access (אבל לא בגירסה 6.5 של שרת SQL), תוכל להשתמש במילת המפתח TOP להגבלת מספר הרשומות המוחזרות. התסריט הבא משתמש במילת המפתח TOP להחזרת 15 הרשומות הראשונות:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL; DATABASE=pubs"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.Open "SELECT TOP 15 * FROM Authors ORDER BY au_lname", Con
```

```
While not RS.EOF
    Response.Write( "au_lname" ) & "<BR>"
    RS.MoveNext
Wend
%>
```

## עדכון רשומות באמצעות Native ADO Methods

בכל התייחסות ל-ADO עד לנקודה זו, הנחנו כי עדכון הרשומות נעשה על ידי הפעלת משפטי SQL, דוגמת המשפטים DELETE ו-UPDATE. אולם, ADO מספק אפשרות נוספת. במקום לשנות רשומות על ידי שימוש ב-SQL, ניתן לשנותן על ידי שימוש ב-Native ADO Methods. באופן כללי, רצוי להמנע מעדכון רשומות באמצעות שיטות ADO, היות והן פחות יעילות מפקודות SQL. עם זאת, ייתכן ובמצבים מסוימים יהיו השיטות של ADO נוחות יותר לשימוש. טבלה 13.5 מפרטת את שיטות ADO לעדכון רשומות.

**טבלה 13.5** שיטות ADO לעדכון רשומות

| שיטה | משמשת ל... |
|---|---|
| AddNew | הוספת רשומה חדשה. |
| CancelUpdate | ביטול כל השינויים שבוצעו ברשומה. כדי שהשיטה תפעל כראוי, יש להשתמש בה לפני קריאה לשיטה Update, ולפני המעבר לרשומה אחרת. |
| Delete | מחיקת הרשומה הנוכחית. |
| Resync | סנכרון מחדש של ערכת הרשומות עם טבלת מסד הנתונים לאחר שינוי, בעת שימוש בסמן מסוג קדימה-בלבד או בסמן סטטי. |
| Update | שמירת כל השינויים שבוצעו ברשומה. שיטה זו נקראת באופן אוטומטי בעת מעבר לרשומה הבאה. |

טבלה 13.6 מפרטת את מאפייני ADO הקשורים לעדכון רשומות :

**טבלה 13.6** מאפייני ADO לעדכון רשומות

| מאפיין | משמש ל... |
|---|---|
| EditMode | החזרת מצב העריכה הנוכחי. הערכים האפשריים למאפיין זה הם adEditNone (לא מתבצעת עריכה), adEditInProgress (הרשימה הנוכחית נערכה, אך לא נשמרה) ו-adEditAdd (רשומה חדשה נוספה, אך לא נשמרה). |
| OriginalValue | בשדה הניתן לעדכון, מכיל מאפיין זה את ערך השדה המקורי ב-Recordset, לפני ששונה. |
| UnderlyingValue | מאפיין זה מכיל את ערך העמודה הנוכחי בטבלת מסד הנתונים. ערך זה עלול להיות שונה מערך המאפיין OriginalValue, במצב בו עודכנה טבלת מסד הנתונים לאחר אחזור Recordset. |

להוספת רשומה חדשה לטבלת מסד הנתונים, ניתן להשתמש בשיטה AddNew, ראה להלן :

```
<-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.LockType = adLockOptimistic
RS.Open "SELECT * FROM Webusers WHERE 1<>1", Con
RS.AddNew
RS( "username" ) = "Andrew Jones"
RS( "password" ) = "won't say"
RS.Update
%>
```

תסריט זה מוסיף רשומה חדשה לטבלת מסד נתונים בשם WebUsers. הרשומה הנוספת מייצגת משתמש ששמו Andrew Jones שסיסמתו won't say.

**הערה:**

הפעלת מחרוזת SQL המכילה גרש עם האובייקט Connection, גורמת לשגיאה (SQL עושה שימוש בתו הגרש לציון תחילה וסיום של מחרוזות). לעומת זאת, שימוש בתו הגרש בתוך שיטות ADO מקומיות לעדכון Recordsets, לא מוביל לשגיאה.

שים לב כי הטבלה נפתחת על ידי שאילתת SQL שאינה מחזירה רשומות. השאילתה מבקשת את כל הרשומות בהן 1<>1. מכיון שלא קיים מצב בו 1 לא שווה לעצמו, אף רשומה לא עונה על דרישות השאילתה, ושום רשומה לא מוחזרת. היות וברצונך רק להוסיף רשומה חדשה, ולא לאחזר רשומות, חוסכת פתיחת Recordset ריק משאבי שרת יקרים.

כמו כן, שים לב כי Recordset נפתח עם סוג נעילה אופטימי (Optimistic Locking). לא ניתן לעדכן רשומות המשתמשות בנעילת ברירת המחדל, שכן נעילת ברירת המחדל היא לקריאה בלבד (Read Only).

כדי למחוק רשומה, ניתן להשתמש בשיטה Delete של ADO, ראה להלן:

```
<-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL; DATABASE=test"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.LockType = adLockOptimistic
RS.Open "SELECT * FROM Webusers WHERE username='Andrew Jones' ", Con
RS.Delete
RS.Update
%>
```

תסריט זה מאחזר מהטבלה WebUsers של Recordset את כל הרשומות בהן שם המשתמש הוא Andrew Jones. לאחר מכן, נקראת השיטה Delete, למחיקת הרשומה הנוכחית. הקריאה לשיטה Delete מוחקת את הרשומה הנוכחית בלבד. במקרה שה-Recordset מכיל מאות רשומות בהן שם המשתמש הוא Andrew Jones, תימחק רשומה אחת בלבד - רשומת Andrew Jones הראשונה.

לבסוף, לעדכון Recordset, יש להציב ערך חדש לשדה. הדוגמה הבאה משמשת להחלפת הסיסמה של Andrew Jones לסיסמה very secret:

```
<-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=test"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.LockType = adLockOptimistic
RS.Open "SELECT * FROM Webusers WHERE userbame='Andrew Jones' ", Con
RS( "password" ) = "very secret"
RS.Update
%>
```

## עדכון אצווה של רשומות

בשימוש בשיטות ADO מקומיות לעדכון רשומות, ברירת המחדל היא עדכון הרשומות במצב העדכון המיידי (Immediate Update Mode). כל שינוי מועבר למסד הנתונים ברגע הקריאה לשיטה Update.

ADO תומך גם בעדכוני אצווה (Batch Updating). כאשר הרשומות מעודכנות כצרור (כקבוצה), מועברים השינויים ברשומות למסד הנתונים כקבוצה. בעת עדכון של מספר רשומות, עדכון במצב אצווה יהיה יעיל יותר מאשר עדכון במצב העדכון המיידי.

כדי לעדכן רשומות במצב אצווה, יש להשתמש בסמן צד-לקוח (Client Side Cursor), סוג הנעילה עדכון אצווה (Batch Update Locking Type), וסמן סטטי או מונע על ידי ערכת מפתחות. התסריט הבא קובע את ערכי מאפיינים אלה, ומוסיף שתי רשומות לטבלה WebUsers במצב של עדכון אצווה:

```
#INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL; DATABASE=test"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.LockType = adLockBatchOptimistic
RS.CursorType = adOpenStatic
RS.CursorLocation = adUseClient
RS.Open "SELECT * FROM Webusers WHERE 1<>1", Con
RS.AddNew
RS( "username" ) = "Bob"
RS( "password" ) = "Secret"
RS.Update
RS.AddNew
RS( "username" ) = "Fred"
RS( "password" ) = "Secret"
RS.Update
RS.UpdateBatch
%>
```

בעת עריכת רשומות במצב עדכון אצווה, ניתן לבטל כל שינוי שבוצע ברשומות לפני הקריאה לשיטה UpdateBatch, על ידי קריאה ל-CancelBatch. השיטה CancelBatch מבטלת את כל השינויים, וה-Recordset מוחזר למצבו הראשוני.

# אחזור רשומות לתוך מערך

במצבים מסוימים, תצטרך לאחזר את רשומות ה-Recordset לתוך מערך. לדוגמה, אם ברצונך לשנות נתונים המיוצגים ב-Recordset מבלי לשנות את רשומות ה-Recordset, תוכל לשנות את הנתונים שבמערך. כדי להקצות את רשומות של ה-Recordset למערך, יש להשתמש בשיטה ()GetRows של האובייקט Recordset. להלן דוגמה:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL; DATABASE=test"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.Open "SELECT MyFirstCol FROM MyTable", Con
MyArray = RS.GetRows()
%>
```

בתסריט זה, מוקצות כל הרשומות ב-Recordset בשם RS למערך בשם MyArray. מערך זה נוצר ומתמלא (Populated) באופן אוטומטי על ידי השיטה ()GetRows. השיטה ()GetRows יוצרת מערך דו-מימדי.

ממלא המקום הראשון מזהה את השדה, והשני מציין את מספר הרשומה.

התסריט הבא משמש להצגת תוכנו של MyArray :

```
<%
FOR I=0 TO UBOUND(MyArray,2)
%>
<BR> First Column: <%=MyArray(0,i)%>
<BR> Second Column: <%=MyArray(1,i)%>
<%
NEXT
%>
```

הפונקציה ()UBOUND של VBScript משמשת לגילוי גודל המימד השני של המערך. הלולאה FOR...NEXT משמשת למעבר על כל איברי המערך בזה אחר זה. המערך מייצג Recordset בעל שתי עמודות. כאשר ערך אינדקס המערך הראשון הוא 0, מיוצגת העמודה הראשונה. כאשר ערכו 1, מיוצגת העמודה השנייה.

# עבודה עם Recordsets שאינם מחוברים למסד הנתונים

בדרך כלל, Recordset משויך (Associated) לחיבור מסד נתונים מסוים, ומייצג נתונים משאילתת SQL. עם זאת, ברמה הבסיסית ביותר, Recordset הוא לא יותר מאשר מכולת מידע. ניתן ליצור Recordset ולאחסן בו כל מידע שהוא. לדוגמה, התסריט הבא יוצר Recordset מבלי לפתוח חיבור (connection) למסד נתונים :

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.Fields.Append "username", adVarchar, 20
RS.Fields.Append "password", adVarchar, 20
RS.Open
RS.AddNew
RS( "username" ) = "Andrew Jones"
RS( "password" ) = "secret"
RS.Update
%>
<BR>Username: <%=RS.( "username" )%>
<BR>Password: <%=RS.( "password" )%>
```

התסריט יוצר Recordset בשם RS. לאחר מכן, מוספים ל-Recordset שני שדות - username ו-password, על ידי שימוש בשיטה Append של האוסף Fields. שני השדות הם מסוג VARCHAR, ובעלי אורך מקסימלי של 20 תווים. לאחר שה-Recordset מכיל מספר שדות, ניתן לפתוח אותו.

**הערה:**

ביצירה של Recordset שאינו מחובר (Disassociated Recordset), ה-Recordset נוצר עם סמן סטטי של צד-לקוח, וסוג נעילה adLockBatchOptimistic.

ניתן להשתמש בכל שיטות ADO המקומיות לעדכון נתונים גם על Recordset חדש. בתסריט, נעשה שימוש בשיטה AddNew להוספת רשומה חדשה ל-Recordset.

דרך נוספת ליצירת Recordset לא מקושר היא על ידי שימוש ב-Recordset רגילה, וניתוקה מחיבור מסד הנתונים. לדוגמה, התסריט הבא מאחזר את כל הרשומות מטבלת מסד הנתונים Authors, ולאחר מכן הוא מנתק את חיבור מסד הנתונים על ידי קביעת המאפיין ActiveConnection ל-Nothing.

```
#INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL; DATABASE=pubs"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.CursorLocation = adUseClient
RS.CursorType = adOpenStatic
RS.LockType = adLockBatchOptimistic
RS.Open "SELECT * FROM Authors", Con
RS.ActiveConnection = Nothing
RS( "au_lname" ) = "Jones"
RS( "password" ) = "Secret"
RS.Update
While not RS.EOF
    Response.Write RS( "au_lname" ) & "<BR>"
    RS.MoveNext
Wend
%>
```

לאחר ניתוק החיבור למסד הנתונים, מעודכן שם משפחתו של הסופר שברשומה הנוכחית ל-Jones. שינוי השם לא משפיע על טבלת מסד הנתונים, היות וכעת ה-Recordset אינו מחובר לטבלה. עם זאת, בעת הצגת כל שמות הסופרים, מוצג ביניהם גם השם החדש.

אחד השימושים המעשיים ל-Recordsets לא מקושרים, הוא הטמנה (Caching) של נתוני מסד נתונים המשתנים לעיתים נדירות. שאילתות למסד הנתונים גוזלות משאבי מערכת רבים. ניתן לאחזר Recordset פעם אחת, להקצותה למשתנה Application ולהשתמש בנתוני Recordset מתוך דפי ה-ASP מבלי לבצע שאילתה נוספת למסד הנתונים.

## יצירת Persistent Recordset

Persistent Recordset היא Recordset שנשמר בכונן הקשיח, או באמצעי אחר. ניתן להשתמש ב-ADO ליצירת Recordset, שמירתו בקובץ, וטעינתו ממנו. בשלב מאוחר יותר, ניתן אף לטעון את ה-Recordset על ידי שימוש ב-URL של אתר באינטרנט.

Recordset ניתן לשמירה בקובץ על ידי שימוש באחת משתי התצורות: תצורת Advanced Data TableGram, ותצורת XML. תצורת Advanced Data TableGram (ADTG) היא תצורת ברירת המחדל. זוהי תצורה בעלת זכויות יוצרים שפותחה ב-Microsoft. XML, לעומתה, היא תצורת תקן פתוח (Open Standard), הנתמכת, בין השאר, על ידי ה-World Wide Web Consortium (ראה http://www.w3.org). להלן שני קבועים המשמשים בתצורות הקבצים:

adPersistADTG: תצורת Advanced Data TableGram. ערכו של קבוע זה הוא 0.

adPersistXML: תצורת XML. ערכו של קבוע זה הוא 1.

כדי לשמור Recordset לקובץ, יש להשתמש בשיטה Save של האובייקט Recordset. הנה דוגמה לשמירת Recordset המייצג את הטבלה Authors לקובץ בתצורת ADTG:

```
#INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL; DATABASE=pubs"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.CursorLocation = adUseClient
RS.CursorType = adOpenStatic
RS.LockType = adLockBatchOptimistic
RS.Open "SELECT * FROM Authors", Con
RS.Save "c:\inetpub\wwwroot\myfile.rst", adPersistADTG
%>
```

**הערה:**

ייתכן ותצטרך להוסיף את הקבועים adPersistADTG ו-adPersistXML לקובץ adovbs.inc, כדי שתוכל להשתמש בהם בדפי ה-ASP. כדי לעשות זאת, הוסף בראש הקובץ את השורות הבאות:

```
Const adPersistADTG = 0
Const adPersistXML = 1
```

תסריט זה שומר Recordset של הסופרים לקובץ בשם myfile.rst. ה-Recordset נשמר בתצורת ADTG (בעצם, השימוש בקבוע adPersistADTG הוא מיותר, היות ותצורת ADTG היא תצורת ברירת המחדל).

בטעינת ערכת הרשומות מהקובץ, ניתן לציין נתיב קובץ מלא, נתיב יחסי, או אפילו URL. הדוגמה הבאה טוענת את Recordset על ידי ציון נתיב הקובץ המלא, ולאחר מכן מציגה את כל הרשומות שב-Recordset:

```
<%
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.Open "c:\myfile.rst"
While not RS.EOF
    Response.Write RS( "au_lname" ) & "<BR>"
    RS.MoveNext
Wend
%>
```

אם קובץ Recordset נמצא באתר אינטרנט, ניתן לטעון את הקובץ על ידי ציון ה-URL. התסריט הבא טוען את אותו Recordset, על ידי ציון URL:

```
<%
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.Open "http://www.somewhere.com/myfile.rst"
While not RS.EOF
    Response.Write RS( "au_lname" ) & "<BR>"
    RS.MoveNext
Wend
%>
```

Persisted Recordsets מספקות אמצעי פשוט להפצת נתוני טבלת מסד הנתונים. ניתן להשתמש ב-Persisted Recordsets להעברת נתונים בין שרתי אינטרנט, או בין שרת אינטרנט ליישום לקוח העושה שימוש ב-ADO.

**הערה:**

השיטה Save מאפשרת שמירה של ערכת נתונים לקובץ. ADO מכיל גם שיטה לשמירת ערכת נתונים למחרוזת: השיטה ()GetString. ניתן להשתמש בשיטה זו ליצירת קבצי CSV מ-Recordsets תוך כדי הפעולה. למידע נוסף, עיין בערך ()GetString שבתיעוד ADO.

# סיכום

בפרק זה, למדת להשתמש באובייקט Recordset של ADO בדפי ASP. למדת לאחזר רשומות ל-Recordset על ידי הפעלת שאילתת SQL. למדת כיצד להשתמש בסוגי סמן ונעילה שונים. לבסוף, למדת את אופן העבודה עם מספר מהשיטות והמאפיינים המתקדמים של ADO.

## פרק 14

# עבודה עם האובייקט Command

**בפרק זה:**

- **יצירת שגרות מאוחסנות (Stored Procedures)**
- **ביצוע שגרות מאוחסנות באמצעות האובייקט Connection**
- **ביצוע שגרות מאוחסנות באמצעות האובייקט Command**
- **אחזור נתוני פרמטרים**

בפרק זה תלמד כיצד להשתמש בשגרות SQL מאוחסנות (SQL Stored Procedures) בתוך דפי ASP. בחלק הראשון, תלמד כיצד ליצור ולהשתמש בשגרות מאוחסנות באמצעות Microsoft SQL Server. בחלק השני, תלמד שיטה קלה לביצוע שגרות מאוחסנות באמצעות האובייקט Connection. לבסוף, תלמד כיצד לשלוט באופן מוחלט על כל המאפיינים של שגרה מאוחסנת באמצעות האובייקט Command.

**הערת העורך:**

המונח Procedure תורגם למילה שגרה. בספר זה המונח שגרה הוא גם תרגום של Function, Routine ו-Subroutine. באופן כללי לכל המונחים האלה יש פירוש אחד והוא ביצוע רצף של משפטי תכנות למטרה אחת.

**הערת העורך:**

מסד נתונים מכיל רשומות. כל **רשומה** מכילה **שדות**. מכיון שאנו מדברים על מסדי נתונים טבלאיים, מתייחסים לאוסף של רשומות כאל **טבלה**, לרשומה בתור **שורה** ולשדה בתור **עמודה**.

# יצירת Stored Procedures

שגרה מאוחסנת (Stored Procedure) של Microsoft SQL Server היא אוסף מהודר (Compiled Collection) של הצהרות SQL המאוחסנות יחדיו כאובייקט בעל שם בודד בתוך מסד נתוני SQL Server. שגרת SQL מאוחסנת יכולה להכיל הצהרת SQL סטנדרטית אחת או יותר, כמו ההצהרות SELECT, INSERT, UPDATE, ו-DELETE המוכרות. כמו כן היא יכולה להכיל הצהרות Transact-SQL.

Transact-SQL היא הרחבה של Microsoft לשפת SQL. Transact-SQL מרחיבה את שפת SQL עד כדי שפת תכנות במובן הרחב של המילה באמצעות הוספת תמיכה במשתנים, בפונקציות ובפקודות בקרת זרימה, כגון תנאים ולולאות WHILE. באמצעות שימוש בשגרות מאוחסנות וב-Transact-SQL, תוכל להעביר תסריטי ASP רבים לשרת מסד הנתונים שלך.

מדוע תרצה לעשות זאת? ובכן, שגרות מאוחסנות מציעות מספר יתרונות:

- אם יש לך ערכה מורכבת של הצהרות SQL לביצוע בדפי ASP רבים, תוכל למקם אותה בשגרה מאוחסנת ולבצע את השגרה במקומה. שיטה זו תקטין את גודלם של דפי ה-ASP שלך ותבטיח ביצוע של אותן הצהרות SQL בכל אחד מהדפים.
- ביצוע סדרת הצהרות SQL בתוך שגרה מאוחסנת יעיל הרבה יותר מאשר ביצוע של אותה סדרת הצהרות בדף ASP. באמצעות מיקומן של ההצהרות בתוך שגרה מאוחסנת, ניתן להימנע מטיולים רבים בין שרת האינטרנט לבין שרת מסד הנתונים.

- ניתן להעביר ערכים הלוך ושוב לשגרה מאוחסנת. שגרות מאוחסנות הן לכן גמישות מאוד. שגרה מאוחסנת אחת יכולה להחזיר מידע שונה מאוד, תלוי בנתונים שהועברו אליה.
- כאשר אתה שולח אוסף הצהרות SQL לשרת מסד הנתונים, יש להעביר כל אחת מההצהרות באופן נפרד. מצד שני, כאשר אתה מבצע שגרה מאוחסנת, מועברת רק הצהרה אחת. לכן באמצעות שימוש בשגרות מאוחסנות, ניתן להפחית את העומס על הרשת.
- ניתן לעצב הרשאות טבלה כך שמשתמשים יוכלו לשנות טבלה רק באמצעות שימוש בשגרות מאוחסנות. כך ניתן לשפר את רמת האבטחה של המידע המאוחסן בטבלאות מסד הנתונים.
- ניתן להפעיל שגרה מאוחסנת אחת מתוך שגרה מאוחסנת אחרת. אסטרטגיה זו מאפשרת לך לבנות שגרות מאוחסנות מורכבות מאוד, משגרות מאוחסנות קטנות יותר. דבר זה אומר גם שניתן לעשות שימוש חוזר באותה שגרה מאוחסנת עבור הרבה משימות תכנות נפרדות.
- שגרות מאוחסנות מהוות יתרון נוסף בכך שהן נמצאות בתוך מסד נתוני SQL SErver ועברו הידור פעם אחת. הפעלתן מהירה ואינה דורשת הידור נוסף (אלא אם בוצע שינוי בשגרה המאוחסנת).

בכל פעם בה עליך לבצע הצהרות SQL בדפי ASP, עליך לשקול בזהירות האם ניתן למקם הצהרות אלו בתוך שגרה מאוחסנת. היתרונות שהוזכרו כאן הינם יתרונות משמעותיים מאוד.

# כיצד ליצור שגרות מאוחסנות

ניתן ליצור שגרה מאוחסנת (Stored Procedure) חדשה באמצעות שימוש ב-Microsoft Query Analyzer (נודעה בעבר בשם I/SQL), ב-Microsoft SQL Server Enterprise Manager, או בתוך תסריט ASP. בכל שלושת המקרים, השגרה המאוחסנת נוצרת באמצעות ההצהרה CREATE PROCEDURE של SQL.

שגרת SQL מאוחסנת יכולה להכיל הצהרה אחת או מאות הצהרות Transact-SQL (גודלה המירבי של שגרה מאוחסנת היא 128 Megabytes). לפניך דוגמה היוצרת שגרה מאוחסנת פשוטה מאוד:

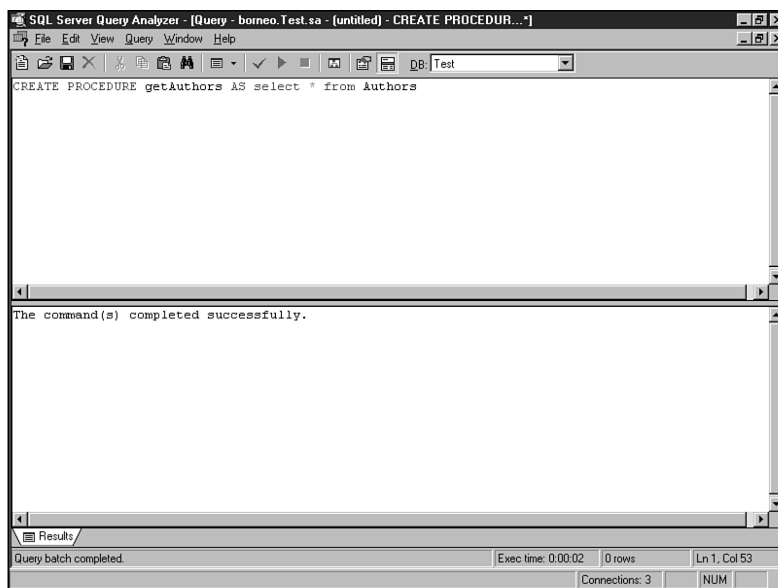
```
CREATE PROCEDURE getAuthors AS select * from Authors
```

אם תבצע הצהרה זו, תיווצר שגרה מאוחסנת בשם getAuthors. שגרה מאוחסנת זו מאחזרת את כל הרשומות מטבלה בשם Authors.

כדי ליצור שגרה זו באמצעות Microsoft Query Analyzer, הפעל תוכנית זו מקבוצת התוכניות של SQL Server 7.0. התחבר לשרת מסד הנתונים שלך ובחר במסד הנתונים pubs. הקלד את ההצהרה הקודמת ולחץ על הלחצן Execute Query (או הקש F5). כעת השגרה נוצרה (ראה תרשים 14.1).

ניתן ליצור שגרה מאוחסנת זו גם באמצעות תסריט ה-ASP הבא:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=pubs"
Con.Execute "CREATE PROCEDURE getAuthors AS select * from Authors"
%>
```




**תרשים 14.1** יצירת שגרה מאוחסנת חדשה באמצעות Microsoft Query Analyzer.

לאחר יצירת השגרה המאוחסנת, תוכל להפעיל אותה ולקבל חזרה את רשימת הסופרים בתוך Microsoft Query Analyzer. זוהי שיטה טובה לבחינת שגרה מאוחסנת לפני הוספתה לתסריט ASP. כדי להפעיל שגרה מאוחסנת ב-Microsoft Query Analyzer, השתמש בהצהרה EXECUTE כמו זאת:

```
EXECUTE getAuthors
```

## הערה:

כאשר שגרה מאוחסנת מופיעה כשורה הראשונה ב- Microsoft Query Analyzer, אינך צריך להשתמש בהצהרה EXECUTE כדי להפעיל את השגרה המאוחסנת - רק הקלד את שמה. אינך יכול להשתמש בקיצור דרך זה אם השגרה מופיעה לאחר הצהרות אחרות. במקרה זה, Query Analyzer פשוט תתבלבל.

השגרה המאוחסנת getAuthors פשוטה מאוד. היא מכילה רק הצהרה אחת המחזירה סדרה של רשומות. שגרה מאוחסנת הופכת מעניינת יותר כאשר היא כוללת משתנים ותנאים לוגיים. הדוגמה הבאה בוחרת רשומות מאחת משתי טבלאות בהסתמך על ערך אקראי:

```
CREATE PROCEDURE ranRows
AS
DECLARE @ranValue INT
SELECT @ranValue = RAND() * 10
IF @ranValue > 5
  SELECT * FROM Authors
ELSE
  SELECT * FROM Titles
```

בדוגמה זו, משתנה בשם ranValue@ מוצהר כטיפוס נתונים מסוג INTEGER (משתנים מקומיים של SQL חייבים להתחיל בתו @). כעת, הפונקציה ()RAND משמשת להשמה של ערך אקראי למשתנה. בהסתמך על ערך אקראי זה, מוחזרות הרשומות מהטבלה Authors או מהטבלה Tables.

**טיפ:**

ניתן לקבל רשימה של כל השגרות המאוחסנות הנכללות במסד הנתונים הנוכחי, ביחד עם רשימה של כל הטבלאות והתצוגות, באמצעות הפעלת שגרת המערכת המאוחסנת sp_help ב- Microsoft Query Analyzer. אם ברצונך לראות את התוכן של שגרה מאוחסנת, הקלד sp_helptext מלווה בשמה של השגרה המאוחסנת.

לאחר שנוצרה, שגרה מאוחסנת ממשיכה להתקיים במסד נתונים מסוים עד להסרתה. כדי למחוק שגרה מאוחסנת, השתמש בהצהרה DROP PROCEDURE של SQL כמו זאת:

```
DROP PROCEDURE getAuthors
```

# יצירת שגרות עם פרמטרים

תוכל להעביר נתונים הלוך ושוב לשגרה מאוחסנת על ידי שימוש בקודי החזרה (Return codes), פרמטרי קלט (Input parameters) ופרמטרי פלט (Output parameters). השיטה הפשוטה ביותר לאחזור נתונים משגרה מאוחסנת היא באמצעות קוד חזרה. קוד חזרה יכול להכיל רק ערך שלם (Integer).

הדוגמה הבאה יוצרת שגרה בשם checkAuthors, אשר עושה שימוש בקוד החזרה כדי לציין האם קיים סופר בשם Green:

```
CREATE PROCEDURE checkAuthor
AS
IF EXISTS( SELECT au_lname FROM Authors
  WHERE au_lname = 'Green' )
  RETURN( 1 )
ELSE
  RETURN( 0 )
```

כאשר מופעלת ההצהרה RETURN בשגרה זו מתבצעת יציאה מהשגרה ומועבר קוד החזרה אל מחוץ לשגרה. השגרה מחזירה את הערך 1 אם שם הסופר Green קיים. אחרת, מחזירה השגרה את הערך 0.

**הערה:**

אל תחזיר את הערך NULL בקוד ההחזרה כיוון שערך NULL מחולל שגיאה. כמו כן עליך לנהוג בזהירות כאשר אתה מחזיר ערכים שליליים כיוון שערכים אלה שמורים עבור קודי שגיאה (1- עד 99-). למידע נוסף, פנה ל- SQL Server Books Online.

תוכל להפעיל שגרה זו ב-Query Analyzer ולראות את קוד ההחזרה. כדי לעשות זאת, בצע את שלושת ההצהרות הבאות:

```
DECLARE @returnCode INT
EXECUTE @returnCode = checkAuthor
SELECT @returnCode
```

השגרה checkAuthors תוכל להיות שימושית הרבה יותר אם היא תוכל לבדוק את קיומו של סופר כלשהו ולא רק את קיומו של Green. כדי לבצע זאת, יש לשנות את השגרה כך שתוכל לקבל פרמטר קלט:

```
CREATE PROCEDURE checkAuthor
(
  @authorName VARCHAR( 40 )
)
AS
IF EXISTS( SELECT au_lname FROM Authors
  WHERE au_lname = @authorName )
  RETURN( 1 )
ELSE
  RETURN( 0 )
```

שגרה זו מקבלת פרמטר קלט בשם authorName@. השגרה בודקת האם קיים בטבלה Authors (סופר) בעל השם אשר הועבר כערך של משתנה הקלט. אם הסופר קיים, מוחזר הערך 1, אחרת, מוחזר הערך 0.

תוכל להפעיל שגרה מאוחסנת זו בתוך Query Analyzer באמצעות שימוש בהצהרה כמו ההצהרה הבאה:

```
EXECUTE checkAuthor 'Frege'
```

הצהרה זו מפעילה את השגרה המאוחסנת checkAuthors עם פרמטר קלט בעל הערך Frege. תוכל כמובן להפעיל את השגרה המאוחסנת עם כל ערך שתחפוץ עבור פרמטר הקלט.

כדי להחזיר נתונים משגרה מאוחסנת, ניתן להשתמש בפרמטר פלט. פרמטר פלט, שלא כמו קוד החזרה, יכול להכיל כל טיפוס נתונים הנתמך על ידי SQL Server. לדוגמה, השגרה המאוחסנת הבאה, getFirstName, מחזירה את שמו הפרטי של סופר כלשהו :

```
CREATE PROCEDURE getFirstName
(
  @lastName VARCHAR( 40 ),
  @firstName VARCHAR( 40 ) OUTPUT
)
AS
SELECT @firstname = au_fname
FROM Authors
WHERE au_lname = @lastname
```

שגרה זו מקבלת פרמטר קלט בשם @lastName ומחזירה פרמטר פלט בשם @firstName. שים לב שפרמטר הפלט נקבע כפרמטר פלט באמצעות מילת המפתח OUTPUT של SQL.

השגרה מקבלת שם משפחה של סופר ומחזירה את שמו הפרטי באמצעות חיפוש בטבלה Authors. אם הסופר אינו קיים בטבלה, פרמטר הפלט מוחזר עם הערך NULL (שלא כמו במקרה של קוד החזרה, כאן לא תיגרם שגיאה).

כדי להפעיל שגרה מאוחסנת זו ב-Query Analyzer, עליך ליצור משתנה אשר יכול לקבל את הערך המוחזר מפרמטר הפלט. תוכל להשתמש בשלוש ההצהרות הבאות כדי להפעיל את השגרה המאוחסנת getFirstName ולהדפיס את פרמטר הפלט שהוחזר :

```
DECLARE @theName VARCHAR( 40 )
EXECUTE getFirstName 'Green', @theName OUTPUT
PRINT @theName
```

שים לב שחובה להשתמש במילת המפתח OUTPUT כאשר קוראים לשגרה המאוחסנת getFirstName. כאשר מבוצעות ההצהרות הקודמות, מודפס על המסך פרמטר הפלט שהוחזר מהשגרה המאוחסנת getFirstName. שגרה מאוחסנת בודדת יכולה להכיל עד 1024 פרמטרים של קלט ופלט. פרמטר יכול להיות מכל טיפוס נתונים, כולל טיפוסי הנתונים TEXT ו-IMAGE.

# ביצוע שגרות מאוחסנות באמצעות האובייקט Connection

הדרך הפשוטה ביותר לביצוע שגרה מאוחסנת בתוך Active Server Page היא על ידי שימוש באובייקט Connection. ביכולתך לבצע שגרת SQL מאוחסנת בדיוק כפי שהיית מבצע פקודות SQL. לדוגמה, תסריט ה-ASP הבא מבצע שגרה מאוחסנת בשם byRoyalty:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=pubs"
Set RS = Con.Execute( "byRoyalty 25" )
While not RS.EOF
  Response.Write RS( "au_id" ) & "<br>"
  RS.MoveNext
Wend
%>
```

השגרה המאוחסנת byRoyalty נכללת בבסיס הנתונים pubs (דוגמת בסיס הנתונים נכללת על ידי SQL Server). השגרה מקבלת משתנה קלט בודד המייצג אחוז תמלוגים מסוים, ומחזירה את רשימת מספרי הזיהוי של הסופרים המקבלים את אותו אחוז תמלוגים. בתסריט זה, רשימת מספרי הזיהוי של הסופרים מוצגת בדפדפן באמצעות ביצוע הלולאה WHILE...WEND.

במקום לבצע שגרה מאוחסנת המחזירה Recordset, ניתן להשתמש באובייקט Connection כדי לבצע שגרה מאוחסנת אשר רק מוסיפה או מעדכנת נתונים בטבלת מסד הנתונים.

התייחס לשגרה המאוחסנת הבאה:

```
CREATE PROCEDURE addAuthor
(
  @fname VARCHAR( 40 ),
  @lname VARCHAR( 40 )
)
AS
INSERT Authors ( au_fname, au_lname )
  VALUES ( @fname, @lname )
```

שגרה מאוחסנת זאת מקבלת כקלט שני פרמטרים, שם פרטי ושם משפחה של סופר, ומכניסה את ערכי הפרמטרים לטבלת הסופרים. כדי לבצע שגרה מאוחסנת זאת ניתן להשתמש בתסריט הבא:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=pubs"
Con.Execute "addAuthor 'Andrew', 'Jones' "
%>
```

תסריט זה מוסיף את שם הסופר Andrew Jones לטבלת הסופרים, באמצעות ביצוע השגרה המאוחסנת addAuthor.

כאשר אתה מבצע שגרת SQL מאוחסנת אינך חייב אפילו לקרוא במפורש לפונקציית ההפעלה של האובייקט Connection. במילים אחרות, ניתן לקרוא לשגרה מאוחסנת כאילו היא פונקציה מקומית של האובייקט Connection. להלן דוגמה לכך:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=pubs"
Con.addAuthor "Andrew", "Jones"
%>
```

תסריט זה מבצע מטלה זהה לזו של התסריט הקודם. אולם, שים לב לכך שהשגרה המאוחסנת addAuthor נקראת כאילו היא פונקציה מקומית של האובייקט Connection.

ביצוע של שגרה מאוחסנת על ידי האובייקט Connection היא שיטה קלה ונוחה. אך לשיטה זאת של קריאה לשגרות מאוחסנות ישנם שני חסרונות משמעותיים. חיסרון ראשון הוא שכאשר אתה מבצע שגרה מאוחסנת על ידי האובייקט Connection אינך יכול להחזיר ערכים או משתני פלט מהתוכנית. האלמנטים היחידים הניתנים להחזרה הם רשומות ממסד הנתונים.

החיסרון השני הוא שביצוע של שגרה מאוחסנת על ידי האובייקט Connection אינה שיטה יעילה כל כך. השרת חייב לבצע עבודה רבה בפיצול הפקודה ובעיבוד הפרמטרים שלה (לדוגמה, ADO חייב לבצע "ניחוש" טוב של סוגי הנתונים של הפרמטרים).

כדי להשתמש בשגרה מאוחסנת בצורה יעילה יותר בדפי ה-ASP שלך, וכדי להיות מסוגל לעבוד עם ערכים ומשתני פלט שמקורם בתוכנית, הנך חייב להשתמש באובייקט Command של ADO. בהמשכו של הפרק נדון ביסודיות באובייקט Command.

# ביצוע שגרות מאוחסנות באמצעות האובייקט Command

האובייקט Command של ADO משמש לייצוג כל פקודה שיכולה להתבצע מול מסד נתונים. בכל אופן, כאשר משתמשים באובייקט Command עם SQL Server, היישום השימושי ביותר של האובייקט הוא בביצוע שגרות SQL מאוחסנות.

ניתן להשתמש באובייקט Command גם כדי לבצע שגרות מאוחסנות המחזירות רשומות נתונים, וגם כדי לבצע שגרות מאוחסנות אשר אינן מחזירות רשומות נתונים. בעזרת האובייקט Command ניתן גם להחזיר קודי החזרה ומשתני פלט משגרה מאוחסנת. נניח לדוגמה שמסד הנתונים שלך מכיל את שגרת SQL המאוחסנת updateHitCount:

```
CREATE PROCEDURE updateHitCount
AS
UPDATE HitCount
  SET Hits = Hits + 1
```

שגרה מאוחסנת זאת מוסיפה בסך הכל 1 לעמודה הנקראת Hits, בטבלה הנקראת HitCount. ניתן להשתמש בשגרה מאוחסנת זאת כדי לעקוב אחרי מספר הכניסות לאתר האינטרנט שלך.

ביכולתך לבצע שגרה מאוחסנת זאת בדפי ASP באמצעות שימוש באובייקט Command. להלן דוגמה לכך:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Set MyCommand = Server.CreateObject( "ADODB.Command" )
Set MyCommand.ActiveConnection = Con
MyCommand.CommandType = adCmdStoredProc
MyCommand.CommandText = "updateHitCount"
MyCommand.Execute
%>
```

תסריט זה יוצר מופע של האובייקט Command באמצעות שימוש בפונקציה CreateObject של האובייקט Server. בשלב הבא, האובייקט Command מקושר עם חיבור (connection) פתוח למסד הנתונים, באמצעות שימוש במאפיין ActiveConnection של Command. המאפיין CommandType משמש כדי לציין כי הפקודה תשמש לביצוע שגרת SQL מאוחסנת. המאפיין CommandText מספק את שם השגרה המאוחסנת במסד הנתונים של SQL, שעומדת להתבצע.

לבסוף, כאשר נקראת פונקציית ההפעלה של האובייקט Command, השגרה המאוחסנת מבוצעת והטבלה hitCount מתעדכנת.

תסריט זה משמש לביצוע שגרה מאוחסנת שאינה מחזירה רשומות נתונים כלל. עם זאת, האובייקט Command יכול לשמש גם לביצוע שגרה מאוחסנת שמחזירה רשומות נתונים. בכדי להחזיר רשומות משגרות מאוחסנות, יש להשתמש באובייקט Command בשילוב עם אובייקט RecordSet. לדוגמה, נניח כי שגרת ה-SQL המאוחסנת getAuthor מאוחסנת בשרת מסד הנתונים שלך. השגרה המאוחסנת הפשוטה והקצרה הבאה, מחזירה את כל הרשומות מטבלת הסופרים:

```
CREATE PROCEDURE getAuthors AS select * from Authors
```

התסריט הבא מבצע שגרה מאוחסנת זאת בפועל, ומציג בדפדפן את רשימת הסופרים המלאה:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=pubs"
Set MyCommand = Server.CreateObject( "ADODB.Command" )
Set MyCommand.ActiveConnection = Con
MyCommand.CommandType = adCmdStoredProc
MyCommand.CommandText = "getAuthors"
Set RS = MyCommand.Execute()
While NOT RS.EOF
  Response.Write RS( "au_fname" ) & "<br>"
  RS.MoveNext
WEND
%>
```

בדוגמה זאת, לאחר שנקראת פונקציית ההפעלה של האובייקט Command, מוחזר אובייקט Recordset. כיון שה-Recordset אינו נוצר באופן מפורש הוא נפתח עם סמן ברירת המחדל קדימה-בלבד, לקריאה-בלבד ועם סוגי נעילה.

ייתכן וכאשר תשתמש באובייקט Command תרצה לפתוח Recordset עם אפשרויות פונקציונליות נוספות. לדוגמה, ייתכן ותרצה להחזיר Recordset שיכול להתעדכן, או שאולי תרצה Recordset שתומך במאפיין paging של ADO (ראה פרק 13). כדי לפתוח Recordset עם Dynamic Cursor או בתבנית נעולה, יש להגדיר מאפיינים אלה במפורש לפני פתיחת האובייקט. להלן דוגמה אשר פותחת Recordset עם סמן דינמי ונעילה אופטימית באמצעות שימוש באובייקט Command:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Set MyCommand = Server.CreateObject( "ADODB.Command" )
Set MyCommand.ActiveConnection = Con
MyCommand.CommandType = adCmdStoredProc
MyCommand.CommandText = "getAuthors"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.CursorType = adOpenDynamic
RS.LockType = adLockOptimistic
RS.Open MyCommand
RS.AddNew
RS( "au_fname" ) = "Andrew"
RS( "au_lname" ) = "Jones"
RS.Update
While NOT RS.EOF
  Response.Write RS( "au_fname" ) & "<br>"
  RS.MoveNext
WEND
%>
```

תסריט זה פותח, באמצעות האובייקט Command, Recordset התומך בנעילה אופטימלית ובסמן דינמי. בשלב הבא, הסופר Andrew Jones מוסף לטבלת מסד הנתונים, וכל רשומות הסופרים מוצגות (השם החדש Andrew יוצג למעשה פעמיים. זאת כיון שהרשומה הנוכחית תחיל ערך זה גם לאחר הוספתה לטבלה, וגם לאחר שהלולאה WHILE...WEND תגיע לרשומה).

## שימוש בקודי חזרה באמצעות האובייקט Command

כאשר אתה מבצע שגרות SQL מאוחסנות, הן תסתיימנה תמיד עם קוד החזרה (Return code). אם הכל עלה יפה, יוחזר הקוד 0. במידה ואירעה תקלה כלשהי, יוחזר קוד שלילי בטווח 1- ל- 99-.

תוכל לגרום להחזרת קודי החזרה שונים משגרה מאוחסנת על ידי שימוש בפקודה Return, ולהשתמש בקודים אלה ככל העולה על רוחך. לדוגמה, השגרה המאוחסנת הבאה מחזירה 1 אם קיים בטבלת הסופרים סופר בשם Green, ו-0 במידה ולא:

```
CREATE PROCEDURE checkAuthor
AS
IF EXISTS( SELECT au_lname FROM Authors
  WHERE au_lname = 'Green' )
  RETURN( 1 )
ELSE
  RETURN( 0 )
```

ניתן להשתמש באובייקט Command ללכידת הקוד המוחזר משגרה מאוחסנת. כדי לעשות זאת, יש ליצור מופע של האובייקט Parameter. להלן דוגמה לכך:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=pubs"
Set MyCommand = Server.CreateObject( "ADODB.Command" )
MyCommand.ActiveConnection = Con
MyCommand.CommandType = adCmdStoredProc
MyCommand.CommandText = "checkAuthor"
Set MyParam = MyCommand.CreateParameter("ReturnCode", adInteger,
   ➥ adParamReturnValue)
MyCommand.Parameters.Append MyParam
MyCommand.Execute
Response.Write MyCommand( "ReturnCode" )
%>
```

דוגמה זאת יוצרת מופע של האובייקט Parameter בשם MyParam כדי ללכוד את קוד ההחזרה מהשגרה המאוחסנת. כל הפרמטרים – לרבות פרמטרי הקלט, הפלט וקודי ההחזרה – מיוצגים ב-ADO בעזרת האובייקט Parameter. יצירת המופע של האובייקט Parameter נעשה על ידי ביצוע הפונקציה CreateParameter של האובייקט Command. כאשר ברצונך להשתמש בקודי החזרה, הינך חייב לספק לפונקציה CreateParameter את הארגומנטים הבאים:

- Name – שם הפרמטר. השם בא לידי שימוש כאשר פונים לפרמטר בתוך תסריט בדפי ASP. בדוגמה הקודמת, משתמשים בשם כאשר מפיקים כפלט את ערך הפרמטר על ידי הפקודה Response.Write.
- Type – טיפוס הנתון שמכיל הפרמטר. בשימוש עם קודי החזרה טיפוס הנתון הוא תמיד adInteger.
- Direction – מציין אם הפרמטר הוא קוד החזרה, קלט או פרמטר פלט. בשימוש עם קודי החזרה הכיוון הוא תמיד adParamReturnValue.

לאחר יצירת המופע של האובייקט Parameter יש להוסיף אובייקט חדש לאוסף הפקודות Parameters. בדוגמה הקודמת, הדבר נעשה באמצעות הקריאה לפונקציה append:

```
MyCommand.Parameters.Append MyParam
```

אם ברצונך לחסוך בקוד, במקום ליצור ולהוסיף את האובייקט Parameter בשתי שורות בתסריט, ביכולתך לשלב את שני השלבים על ידי שימוש בפקודות הבאות:

```
MyCommand.Parameters.Append
  ↳ MyCommand.CreateParameter( "returnCode", adInteger,
  ↳ adParamReturnValue)
```

פקודה זו משלבת הן את יצירת האובייקט Parameter, והן הוספה של אובייקט Parameter חדש לאוסף הפקודות Parameters.

ניתן לקבל ערך מכל שגרה מאוחסנת. ניתן לקבל ערך אפילו משגרה מאוחסנת המחזירה רשומות נתונים. הבט בשגרה המאוחסנת הבאה:

```
CREATE PROCEDURE checkGetAuthors
AS
SELECT * FROM Authors
IF EXISTS( SELECT au_lname FROM Authors
  WHERE au_lname = 'Green' )
  RETURN( 1 )
ELSE
  RETURN( 0 )
```

שגרה מאוחסנת זו מבצעת שני דברים. היא מחזירה את כל הרשומות מהטבלה Authors, אולם בנוסף היא מחזירה ערך מסוים. תוכל להציג את הרשומות ואת הערך אשר התקבלו כפלט מהשגרה המאוחסנת באמצעות שימוש בתסריט הבא:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=pubs"
SET MyCommand = Server.CreateObject( "ADODB.Command" )
MyCommand.ActiveConnection = Con
MyCommand.CommandType = adCmdStoredProc
MyCommand.CommandText = "checkGetAuthors"
Set MyParam = MyCommand.CreateParameter( "returnCode",
   ↳ adInteger, adParamReturnValue )
MyCommand.Parameters.Append MyParam
SET RS = MyCommand.Execute()
WHILE NOT RS.EOF
  Response.Write RS( "au_fname" )
  RS.MoveNext
WEND
RS.Close
Response.Write MyCommand( "returnCode" )
%>
```

קיימת מגבלה אחת חשובה מאוד כאשר מאחזרים ערכים באמצעות שגרות המחזירות רשומות. לפני שתוכל לגשת לערך, עליך לסגור את אובייקט Recordset המכיל את הרשומות. כלומר, ניתן לגשת לערך שהחזירה השגרה רק לאחר סיום הצגת הרשומות כולן. מגבלה זו אינה נוחה במיוחד. לדוגמה, היה זה נחמד אילו היתה שיטה להחזיר כערך את מספר הרשומות, ואז להציג את כל הרשומות. בכל מקרה, שיטה כזו אינה קיימת.

# שימוש בפרמטרי קלט באמצעות האובייקט Command

כדי להעביר ערכים לשגרה מאוחסנת, עליך להשתמש בפרמטר קלט (Input parameter). לדוגמה, השגרה המאוחסנת הבאה מקבלת פרמטר קלט בשם firstLetter@ המשמש כדי לאחזר את כל הסופרים ששם משפחתם מתחיל באות מסוימת:

```
CREATE PROCEDURE getAuthorsByLastName
(
  @firstLetter VARCHAR( 1 )
)
AS
SELECT * FROM Authors
WHERE au_lname LIKE @firstLetter + '%'
```

נניח שברצונך להציג את כל הסופרים ששמם מתחיל באות G מטבלת מסד הנתונים Authors בתוך דף ASP. תוכל לבצע זאת באמצעות התסריט הבא:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=pubs"
SET MyCommand = Server.CreateObject( "ADODB.Command" )
MyCommand.ActiveConnection = Con
MyCommand.CommandType = adCmdStoredProc
MyCommand.CommandText = "getAuthorsByLastName"
MyCommand.Parameters.Append MyCommand.CreateParameter
  ➥ ("ReturnCode", adInteger, adParamReturnValue)
MyCommand.Parameters.Append MyCommand.CreateParameter
  ➥ ("firstLetter", adVarchar, adParamInput, 1)
MyCommand.Parameters("firstLetter") = "g"
SET RS= MyCommand.Execute()
WHILE NOT RS.EOF
 REsponse.WRite RS( "au_lname" ) & "<br>"
 RS.MoveNext
WEND
%>
```

תסריט זה יוצר שני פרמטרים: פרמטר קוד ההחזרה ופרמטר קלט. פרמטר הקלט נוצר באמצעות שימוש בשיטה CreateParameter של האובייקט Command. טבלה 14.1 מציגה את הארגומנטים המשמשים את השיטה כאשר עובדים עם פרמטרי קלט.

**טבלה 14.1** ארגומנטים של השיטה CreateParameter עבור יצירת פרמטר קלט

| ארגומנט | שימוש |
|---|---|
| Name | שמו של הפרמטר. השם משמש בעת השמת ערך לפרמטר הקלט לפני הפעלת הפקודה. |
| Type | טיפוס הנתונים של הפרמטר. בתסריט הקודם, מצוין פרמטר מסוג VARCHAR באמצעות שימוש בקבוע adVarChar. טיפוסי נתונים שכיחים נוספים הם adInteger עבור מספרים שלמים (Integer) ו-adLongVarChar עבור טקסט. |
| Direction | מציין שהפרמטר הינו פרמטר קלט. בעת שימוש עם פרמטרי קלט, הכיוון הוא תמיד adParamInput. |
| Size | כאשר יוצרים משתנה מטיפוס נתונים בעל אורך משתנה, כמו VARCHAR, ארגומנט זה מספק את גודלו המקסימלי של הפרמטר. חובה לציין גודל בעת שימוש בפרמטר מסוג VARCHAR, אך לא בעת שימוש בפרמטרים מסוג INTEGER. |

# שימוש בפרמטרי פלט באמצעות האובייקט Command

כאשר אתה מאחזר רשומה בודדת מטבלת מסד נתונים, או כאשר אתה מחשב ערך בודד, עליך לעשות שימוש בשגרה מאוחסנת (Stored procedure) המחזירה פרמטר פלט (Output parameter). שימוש בפרמטר פלט הוא תמיד יעיל יותר מאשר שימוש באובייקט Recordset כדי לאחזר רשומה בודדת מטבלת מסד נתונים.

לדוגמה, נניח שעליך לאחזר מטבלה כלשהי כתובת דואר אלקטרוני של אדם מסוים כאשר כל המידע שבידך הוא שמו הפרטי של אותו אדם. אולי תתפתה לפתוח Recordset עם שאילתת SQL כדי למצוא את כתובת הדואר האלקטרוני. אל תעשה זאת! במקום זאת, השתמש בשגרה המאוחסנת הבאה:

```
CREATE PROCEDURE getEmail
(
  @username VARCHAR( 20 ),
  @email VARCHAR( 100 ) OUTPUT
)
AS
SELECT @email = email
FROM Webusers
WHERE username = @username
```

כדי להשתמש בשגרה מאוחסנת זו בדף ASP, עליך לאחזר את ערכו של פרמטר הפלט לתוך מופע של האובייקט Parameter של ADO. התסריט הבא ממחיש כיצד לעבוד עם פרמטר פלט:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=pubs"
SET MyCommand = Server.CreateObject( "ADODB.Command" )
MyCommand.ActiveConnection = Con
MyCommand.CommandType = adCmdStoredProc
MyCommand.CommandText = "getEmail"
MyCommand.Parameters.Append MyCommand.CreateParameter
  ➥ ( "ReturnCode", adInteger, adParamReturnValue)
MyCommand.Parameters.Append MyCommand.CreateParameter
  ➥ ( "username", adVarchar, _adParamInput, 20)
MyCommand.Parameters( "username" ) = "Andrew Jones"
MyCommand.Parameters.Append MyCommand.CreateParameter
  ➥ ( "email", adVarchar, adParamOutput, 100)
MyCommand.Execute
Response.Write MyCommand( "email" )
%>
```

שגרה זו מעבירה פרמטר קלט בשם username לשגרה המאוחסנת ומאחזרת את כתובת הדואר האלקטרוני מפרמטר הפלט email. פרמטר פלט נוצר באופן זהה לזה של פרמטר קלט, באמצעות שימוש בשיטה CreateParameter. טבלה 14.2 מציגה את הארגומנטים המשמשים את השיטה כאשר עובדים עם פרמטרי פלט.

**טבלה 14.2** ארגומנטים של השיטה CreateParameter עבור יצירת פרמטר פלט

| ארגומנט | שימוש |
|---|---|
| Name | שמו של הפרמטר. השם משמש בעת אחזור הערך מפרמטר הפלט לאחר ביצוע הפקודה. |
| Type | טיפוס הנתונים של הפרמטר. בתסריט הקודם, משמש פרמטר מסוג VARCHAR עבור הפרמטר email באמצעות שימוש בקבוע adVarChar. טיפוסי נתונים שכיחים נוספים הם adInteger עבור מספרים שלמים (Integer) ו-adLongVarChar עבור טקסט. |
| Direction | מציין שהפרמטר הינו פרמטר פלט. בעת שימוש עם פרמטרי פלט, הכיוון הוא תמיד adParamOutput. |
| Size | כאשר יוצרים משתנה מטיפוס נתונים בעל אורך משתנה, כמו VARCHAR, ארגומנט זה מספק את גודלו המקסימלי של הפרמטר. חובה לציין גודל בעת שימוש בפרמטר מסוג VARCHAR, אך לא בעת שימוש בפרמטרים מסוג INTEGER. |

באותו האופן בו אתה יכול להשתמש בפרמטר קוד החזרה עם שגרה מאוחסנת למעבר שורה של SQL, תוכל גם להשתמש בפרמטר פלט. אם שגרה מאוחסנת מחזירה רשומות, אתה חייב לסגור את אובייקט Recordset לפני שתוכל לגשת לפרמטר הפלט. אחרת, תאחזר ערך ריק.

# פרמטרי פלט ועמודות זיהוי

Microsoft SQL Server תומך בטבלאות המכילות עמודות זיהוי (Identity columns). עמודת זיהוי היא טיפוס מיוחד של עמודת טבלה אשר מובטח עבורה ערך ייחודי. עמודת זיהוי משמשת בדרך כלל לזיהוי ייחודי של שורת נתונים עבור אחזורה המאוחר יותר.

לדוגמה, ההצהרה הבאה יוצרת טבלה המכילה עמודת זיהוי בשם au_id:

```
CREATE TABLE Authors (au_id INT IDENTITY, au_name VARCHAR( 50 ) )
```

בכל פעם בה מוכנסת רשומה חדשה לתוך טבלה זו, העמודה au_id תגדל באופן אוטומטי כך שתכיל ערך חדש.

כאשר אתה עובד עם טבלה המכילה עמודת זיהוי, אולי תצטרך להכניס גם את הרשומה החדשה וגם לאחזר את ערכה של עמודת הזיהוי עבור הרשומה החדשה. ביצוע משימה זו באמצעות שימוש באובייקט Recordset ובאובייקט Connection בלבד היא משימה קשה. לאחר הכנסת הרשומה החדשה, קשה לזהותה שנית כך שתוכל לאחזר את עמודת הזיהוי שלה (זו הסיבה לצורך הראשוני בעמודה).

למרבה המזל, הכנסתה של רשומה ואחזור עמודת הזיהוי שלה היא משימה קלה כאשר משתמשים בשגרות מאוחסנות ובאובייקט Command. השגרה המאוחסנת (Stored procedure) הבאה, הנקראת addAuthor, מוסיפה שם של סופר חדש ומאחזרת את ערכה של עמודת הזיהוי (Identity column) עבור השורה החדשה:

```
CREATE PROCEDURE addAuthor
(
  @AuthorName VARCHAR( 50 ),
  @id INT OUTPUT
)
AS
INSERT Authors ( au_name )
  VALUES ( @AuthorName )

SELECT @id = @@IDENTITY
```

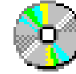
שגרה מאוחסנת זו עושה שימוש בפרמטר קלט עבור שמו של הסופר ובפרמטר פלט עבור ערכה של עמודת הזיהוי לשורה החדשה. המשתנה הגלובלי @@IDENTITY מכיל את ערכה של עמודת הזיהוי בכל פעם בה מוכנסת לטבלה שורה חדשה (המשתנה שומר את ערך הזיהוי עד להכנסתה של שורה חדשה).

התסריט הבא מכניס שם של סופר חדש לתוך הטבלה Authors ומציג את ערכה של עמודת הזיהוי עבור השורה החדשה:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
SET MyCommand = Server.CreateObject( "ADODB.Command" )
MyCommand.ActiveConnection = Con
MyCommand.CommandType = adCmdStoredProc
MyCommand.CommandText = "addAuthor"
MyCommand.Parameters.Append MyCommand.CreateParameter
  ↳ ( "ReturnCode", adInteger, adParamReturnValue)
MyCommand.Parameters.Append MyCommand.CreateParameter
  ↳ ( "AuthorName", adVarchar, adParamInput, 50)
MyCommand.Parameters( "AuthorName" ) = "Andrew Jones"
MyCommand.Parameters.Append MyCommand.CreateParameter
  ↳ ( "id", adInteger, adParamOutput )
MyCommand.Execute
Response.Write MyCommand( "id" )
%>
```

# אחזור נתוני פרמטרים

ייתכן ותרצה להשתמש בשגרה מאוחסנת (Stored procedure) ללא ידיעת הפרמטרים הדרושים עבור השגרה. לדוגמה, ייתכן ואינך יודע את טיפוס הנתונים או את הגודל של הפרמטרים. כיצד תוכל בכל זאת לדעת נתונים אלה?



תוכל לאחזר מידע על פרמטרים של שגרה מאוחסנת באמצעות שימוש בתסריט המוצג בתדפיס 14.1 (ראה תרשים 14.2). תדפיס 14.1 נכלל בתקליטור תחת השם displayParams.asp.

Parameter Information - Microsoft Internet Explorer

Parameter Information

| Parameter Name | Datatype | Direction | Size |
|---|---|---|---|
| RETURN_VALUE | 3 | 4 | 0 |
| @AuthorName | 200 | 1 | 50 |
| @id | 3 | 3 | 0 |

**תרשים 14.2** הצגת מידע על פרמטרים

**תדפיס 14.1** displayParams.asp

```
<!-- #INCLUDE VIRTUAL="ADOVBS.inc" -->
<%
Set MyConn=Server.CreateObject("ADODB.Connection")
Set MyCommand=Server.CreateObject("ADODB.Command")
MyConn.Open "FILE NAME=c:\myDataLink.UDL"
Set MyCommand.ActiveConnection=MyConn
MyCommand.CommandType=adCMDStoredProc
MyCommand.CommandText="sp_myproc"
MyCommand.Parameters.Refresh
%>
<HTML>
<HEAD><TITLE>Parameter Information</TITLE></HEAD>
<BODY>
<TABLE BORDER=1>
<CAPTION>Parameter Information</CAPTION>
```

```
<TR>
 <TH>Parameter Name</TH>
 <TH>Datatype</TH>
 <TH>Direction</TH>
 <TH>Size</TH>
</TR>
<%For Each thing in MyCommand.Parameters %>
<TR>
 <TD><%=thing.name %></TD>
 <TD><%=thing.type %></TD>
 <TD><%=thing.direction %></TD>
 <TD><%=thing.size %></TD>
</TR>
<%
Next
MyConn.Close
%>
</TABLE>
</BODY>
</HTML>
```

דוגמה זו מציגה את כל נתוני הפרמטרים עבור השגרה sp_myproc. השם, טיפוס הנתונים, הכיוון והגודל של כל הפרמטרים מוצגים בטבלה (הכיוון מציין האם הפרמטר הינו פרמטר קלט, פרמטר פלט או קוד החזרה). כדי להציג מידע על שגרה מאוחסנת שונה, פשוט החלף את שם השגרה החדשה עם sp_myproc.

ההצהרה העיקרית בדוגמה זו היא ההצהרה MyCommand.Parameters.Refresh. כאשר מופעלת הצהרה זו, נתוני הפרמטרים של השגרה המאוחסנת מאוחזרים ממסד הנתונים.

תסריט זה אינו מחזיר קבועים. לחילופין, הוא מחזיר ערכי רשומות. כדי לפרש את הערכים המוחזרים על ידי התסריט, עליך לבחון את קובץ include בשם ADOVBS. בקובץ זה, הערכים המספריים של הרשומות מקושרים עם הקבועים המדויקים.

# סיכום

בפרק זה, למדת כיצד לעבוד עם שגרות SQL מאוחסנות בתוך דפי ASP. למדת כיצד ליצור ולהשתמש בשגרות מאוחסנות עם Microsoft SQL Server. בנוסף, למדת שיטה קלה להפעלת שגרות מאוחסנות בתוך דפי ASP באמצעות שימוש באובייקט Connection. לבסוף, למדת כיצד להשתמש בשגרות מאוחסנות עם פרמטרים בדפי ASP באמצעות שימוש שיטות ובמאפיינים של האובייקט Command.

## פרק 15

# עבודה עם Index Server וחיפוש בטקסט-מלא

**בפרק זה**

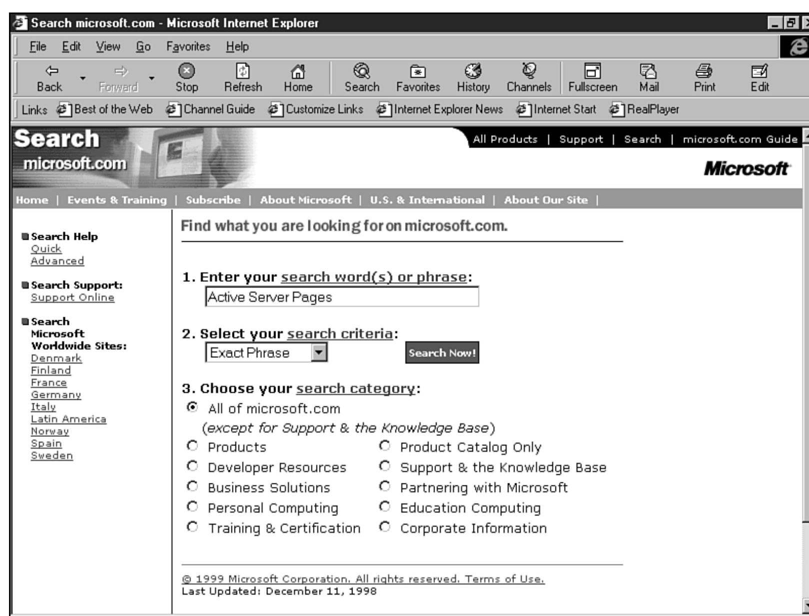
- **שימוש ב-ADO עם Microsoft Index Server**
- **שימוש ב-ADO עם SQL 7.0 Full-Text Search**

בשלושת הפרקים הקודמים, למדת כיצד להשתמש ב- ActiveX Data Objects (ADO) כדי לעבוד עם מסדי נתונים. בכל אופן, Microsoft תכננה את ADO (ואת שותפו השקט OLE DB) לעבודה עם כל סוג של מקור נתונים, לא רק עם מסד נתונים. בפרק זה תלמד כיצד להשתמש ב-ADO כדי לבצע שאילתה בשרת חיפוש.

פרק זה מתאר למעשה כיצד להשתמש ב-ADO עם שני שרתי חיפוש שונים. ראשית, תלמד כיצד להשתמש ב-ADO עם Microsoft Index Server. Microsoft Index Server כלול ב-NT Option Pack, והוא משמש למיפתוח (Indexing) של מסמכים סטטיים. לדוגמה, תוכל להשתמש ב-Index Server כדי למפתח (To Index) ולבצע חיפושים של דפי HTML, מסמכי Microsoft Word, וגיליונות אלקטרוניים של Excel.

חלקו השני של הפרק מתאר כיצד להשתמש ב-ADO עם SQL 7.0 Full-Text Search. SQL 7.0 Full-Text Search מוכלל ב-Microsoft SQL 7.0. הוא משמש למיפתוח ולביצוע שאילתות מול נתונים המאוחסנים בטבלת מסד נתונים.

תוכל להשתמש בכל אחד משני שרתים אלה ליצירת דף חיפוש סטנדרטי לאתר האינטרנט שלך. לדוגמה, אם תבקר בדף http://search.microsoft.com, תראה את דף החיפוש המוצג בתרשים 15.1. לאחר שתקרא פרק זה, תוכל ליישם את אותו סוג חיפוש באמצעות שימוש ב-ASP וב-ADO.




**תרשים 15.1** דף החיפוש של Microsoft

## הערה:

Microsoft מציעה שרת חיפוש שלישי אשר ארוז עם Microsoft Site Server. שרת חיפוש זה לא מתואר בפרק זה. למידע נוסף, בקר באתר האינטרנט של Microsoft Site Server בכתובת http://www.microsoft.com/siteserver.

# שימוש ב-ADO עם Microsoft Index Server

Microsoft Index Server ייצור אינדקס למסמכים סטטיים באופן אוטומטי, כולל דפי HTML, מסמכי Microsoft Word, קבצי PDF, גיליונות אלקטרוניים של Excel, מצגות PowerPoint וקבצי טקסט פשוטים. תוכל להשתמש ב-Microsoft Index Server למיפתוח (ליצירת אינדקס) כל הקבצים בשרת האינטרנט שלך וכדי לאפשר למשתמשים לחפש מסמכים מסוימים באמצעות שימוש בטופס חיפוש.

**הערת העורך:**

הפועל **to Index** תורגם **למפתח** (מהמילה מפתח). לפעמים מונח זה תורגם לייצר אינדקס או לעדכן מחדש אינדקס.

מסד נתונים מכיל רשומות. כל **רשומה** מכילה **שדות**. מכיון שאנו מדברים על מסדי נתונים טבלאיים, מתייחסים לאוסף של רשומות כאל **טבלה**, לרשומה בתור **שורה** ולשדה בתור **עמודה**.

Index Server ימפתח (יצור אינדקס) גם של דפי ASP. אך הוא ימפתח (יעשה אינדקס) רק תוכן סטטי המאוחסן בדף ASP. במילים אחרות, המיפתוח יהיה של המקור אך לא של תוכנו המפוענח של הדף. Index Server לא ימפתח, לדוגמה, שום מידע שדף ASP יאחזר מתוך טבלה.

Microsoft Index Server כלול ב-Windows NT 4.0 Option Pack. היתה בידך האפשרות להתקין אותו כאשר התקנת את IIS. כדי לבדוק האם Index Server מותקן במחשב שלך בחר ב-Start (התחל), Programs (תוכניות), Windows NT 4.0 Option Pack, ובדוק האם קיימת תיקיה בשם Microsoft Index Server. אם התיקיה אינה קיימת, תוכל להתקין את Index Server באמצעות הפעלת תוכנת ההתקנה של Windows NT 4.0 Option Pack.

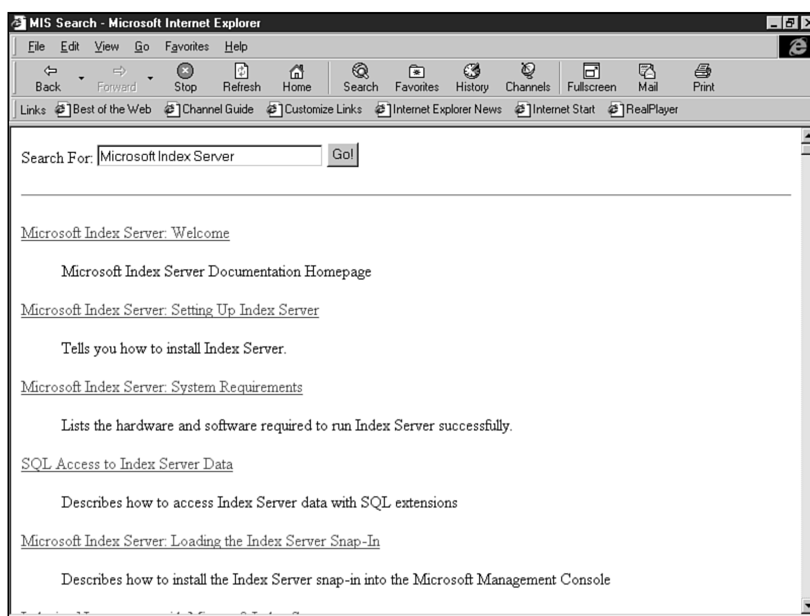
לאחר ש-Index Server הותקן, הוא יעבוד בקביעות ברקע כדי למפתח את כל המסמכים הממוקמים בספריות הווירטואליות בשרת האינטרנט שלך. כברירת מחדל, בכל פעם שתוסיף ספריה וירטואלית, Index Server יאתר את הימצאותה באופן אוטומטי ויתחיל לבצע מיפתוח (אינקדס) של המסמכים הנמצאים בתוכה.

**טיפ:**

ניתן לאלץ את Index Server לבצע מיפתוח (אינקדס) של ספריה בכל זמן שהוא באמצעות שימוש ב- Index Server Manager. בחר את שם הספריה ובחר באפשרות Action, Rescan.

## יצירת טופס חיפוש פשוט

בפרק זה תלמד כיצד להשתמש ב-ADO עם Index Server ליצירת דף חיפוש פשוט עבור אתר האינטרנט שלך. דף החיפוש מאחזר את כל המסמכים בעלי ההתאמה לביטוי חיפוש שהוקלד בטופס HTML (ראה תרשים 15.2). כמו כן, הוא מציג רשימת תקצירים (Abstracts) עבור כל המסמכים המתאימים. הקוד השלם עבור דף החיפוש נכלל בתדפיס 15.1, אשר יוצג מאוחר יותר, והוא נמצא בתקליטור תחת השם MISsearch.asp.




**תרשים 15.2** דף החיפוש של Index Server

דף החיפוש מקים קשר עם Microsoft Index Server באמצעות שימוש ב-OLE DB Provider for Microsoft Index Server. ההצהרה הבאה משמשת להקמת הקשר:

```
Con.Open "PROVIDER=msidxs"
```

לאחר הקמת הקשר, מופעלת ההצהרה הבאה לביצוע החיפוש עצמו:

```
SELECT  Characterization, DocTitle, VPath
FROM  SCOPE() WHERE
FREETEXT(Contents, '" & search & "') > 0
ORDER BY Rank DESC
```

לשאילתה זו ארבעה חלקים. חלק ראשון, ההצהרה SELECT משמשת לאחזור מספר מאפיינים של המסמכים המוחזרים על ידי השאילתה, בדוגמה זו, תקציר (Characterization), כותרת (DocTitle) והנתיב הווירטואלי המלא (Vpath) עבור כל מסמך שאוחזר. תלמד יותר על מאפיינים אלה בסעיף מאוחר יותר בפרק, **מאפיינים ניתנים לחיפוש**.

חלק שני, המשפט FROM משמש לציון טווח החיפוש. במקרה זה נעשה החיפוש בכל מסמך המאוחסן בכל תיקיה וירטואלית המאפשרת מיפתוח. כברירת מחדל, זה אומר שהחיפוש יתבצע בכל הספריות הווירטואליות בשרת האינטרנט שלך.

חלק שלישי, המשפט WHERE, משתמש במילת המפתח FREETEXT (Predicate FREETEXT) כדי להחזיר רק את הרשומות המתאימות לביטוי החיפוש אשר הוקלד בטופס ה-HTML. המאפיין Contents מייצג את תוכנו של המסמך.

לבסוף, המשפט ORDER BY משמש למיון תוצאות השאילתה באמצעות רמת הקשר לביטוי החיפוש. התאמות טובות יותר נמנות בהתחלה. המאפיין Rank מכיל ערך בין 0 ל-1000 המציין את רמת ההתאמה של התוצאה לביטוי החיפוש (מספר גבוה יותר מציין התאמה טובה יותר).

לאחר ביצוע השאילתה, מוחזרות התוצאות באובייקט ערכת רשומות (Recordset) רגיל של ADO. הלולאה WHILE...WEND משמשת להצגת כל הרשומות בערכת הרשומות, אחת אחרי השנייה.

**תדפיס 15.1** MISsearch.asp

```
<html>
<head><title>MIS Search</title></head>
<body>
<form method="post" action="MISsearch.asp">
Search For: <input name="search" size=30>
<input type="submit" value="Go!">
</form>
<hr>
<%
search = TRIM( Request( "search" ) )
if search <> "" then
  search = Replace( search, "'", "" )
  set con=server.createobject( "adodb.Connection" )
  con.Open "provider=msidxs"
  mySQL = "SELECT  Characterization, " &
    ↳ " DocTitle, VPATH " &
    ↳ "FROM  SCOPE() " &
    ↳ "WHERE FREETEXT(Contents, '" & search & "') > 0 " &
    ↳ "ORDER BY Rank DESC"
  Set RS = Con.Execute( mySQL )
  if RS.EOF then
    Response.Write "No matches found!"
```

```
  else
    WHILE NOT RS.EOF
    %>
    <p><a href="<%=RS( "VPath" )%>"><%=RS( "DocTitle" )%></a>
    <blockquote>
    <%=Server.HTMLEncode( RS( "Characterization" ) )%>
    </blockquote>
    <%
    RS.MoveNext
    WEND
  end if
end if
%>
```

# הגבלת טווח החיפוש

בוודאי תרצה להגביל חיפושים שונים לקבוצות מסמכים שונות. לדוגמה, אתר האינטרנט שלך יכול לכלול אזור אחד עבור שרות לקוחות ואזור אחר עבור תיאור המוצרים. אולי תרצה דף חיפוש שונה עבור כל אחד מהאזורים.

ניתן להגביל טווח של שאילתה באמצעות שימוש בפונקציה SCOPE. הפונקציה SCOPE מקבלת שני ארגומנטים. הארגומנט הראשון, Traversal Type, מציין האם החיפוש יתבצע גם בתת-הספריות של ספריית החיפוש. Traversal Type יכול לקבל את הערך Deep Traversal (החיפוש מתבצע בתת-ספריות) ו-Shallow Traversal (החיפוש אינו מתבצע בתת-ספריות). כברירת מחדל, מתבצע החיפוש בכל תת-הספריות.

הארגומנט השני, Path, מכיל רשימה של כל תת-הספריות הווירטואליות והפיסיות אשר בהן יתבצע החיפוש. ניתן לרשום ספריות רבות באמצעות הפרדת נתיב כל ספריה עם פסיק. כברירת מחדל, אם לא תרשום אף ספריה, החיפוש יתבצע בכל הספריות.

הדוגמה הבאה מגבילה את טווח שאילתת החיפוש לספריה וירטואלית בשם ProductInfo. מילות המפתח SHALLOW TRAVERSAL משמשות כדי לציין שהחיפוש לא יתבצע בתת-הספריות של הספריה ProductInfo:

```
<%
Set Con=server.CreateObject( "adodb.Connection" )
Con.Open "PROVIDER=msidxs"
mySQL = "SELECT DocTitle FROM " &
  ↳ "SCOPE( 'SHALLOW TRAVERSAL OF ""/ProductInfo"" ' )  " &
  ↳ "WHERE FREETEXT( Contents, 'PART #THX1138' )"
Set RS = Con.Execute( mySQL )
While NOT RS.EOF
%>
<p>Title: <%=RS( "DocTitle" )%>
<%
RS.MoveNext
Wend
%>
```

# מאפיינים ניתנים לחיפוש

Microsoft Index Server עוקב באופן אוטומטי אחר מספר מאפיינים של המסמכים שמופתחו. עבור סוגים שונים של מסמכים - לדוגמה, מסמכי Microsoft Word או מסמכי HTML - הוא מבצע מעקב אחר מאפיינים שונים. תוכל להשתמש במאפיינים אלה בשאילתות החיפוש שלך באותו אופן בו היית משתמש בשמה של עמודת טבלת מסד נתונים. קיים מספר רב של מאפיינים משאוכל למנות כאן (קרא את התיעוד ל-Microsoft Index Server כדי לראות את הרשימה המלאה), אך אציין מספר מאפיינים מהשימושיים ביותר:

- Access (גישה) - מציין את התאריך והזמן האחרון בו התבצעה גישה למסמך.
- Characterization (תיאור) - מכיל סיכום תמציתי של המסמך. Index Server יוצר סיכומים אלה באופן אוטומטי תוך כדי מיפתוח המסמכים.
- Contents (תוכן) - תוכן המסמך. ניתן להשתמש במאפיין זה במשפט WHERE, אך אי אפשר להחזירו בתוך רשימת SELECT.
- Create (יצירה) - מציין את התאריך והזמן בו נוצר המסמך.
- Directory (ספריה) - הנתיב הפיסי של הספריה בה ממוקם המסמך.
- DocAuthor - שמו של כותב המסמך.
- DocTitle - כותרת המסמך.
- FileName - שם קובץ המסמך.
- HitCount - מספר המילים המוכלים במסמך אשר מתאימים לשאילתת החיפוש.
- Path (נתיב) - הנתיב הפיסי המלא של המסמך.
- Rank (דרגה) - דרגת הקשר של המסמך ביחס לשאילתת החיפוש. יכול לקבל ערכים בין 0 ל-1000 (המספר הגבוה יותר מציין התאמה טובה יותר).
- Size (גודל) - גודלו של המסמך בבתים (Bytes).
- VPath - הנתיב הווירטואלי של המסמך.
- Write - מציין את התאריך והזמן האחרון בו נכתב המסמך.

נניח לדוגמה, שברצונך לאחזר את רשימת המסמכים אשר גודלם אינו עולה על 100 בתים. תוכל לבצע שאילתה זו באמצעות שימוש במאפיין Size:

```
<%
Set con=server.createobject( "adodb.Connection" )
con.Open "provider=msidxs"
mySQL = "SELECT DocTitle, Size " &
 ↳ "FROM  SCOPE()  WHERE " &
 ↳ "Size < 100"
Set RS = Con.Execute( mySQL )
While NOT RS.EOF
%>
```

```
<p>Title: <%=RS( "DocTitle" )%>
<br>Size: <%=RS( "Size" )%>
<%
RS.MoveNext
Wend
%>
```

# יצירת מאפיינים מותאמים אישית ניתנים לחיפוש

אחת התכונות המעניינות ביותר של Index Server היא תמיכתו בהגדרות של **מאפיינים מותאמים אישית** (Custom Properties). לדוגמה, ניתן להוסיף מאפיין מותאם אישית לקבוצה של מסמכי HTML ואז לבצע שאילתות אשר עושות שימוש במאפיין זה כאשר הן מבצעות חיפוש במסמכי ה-HTML.

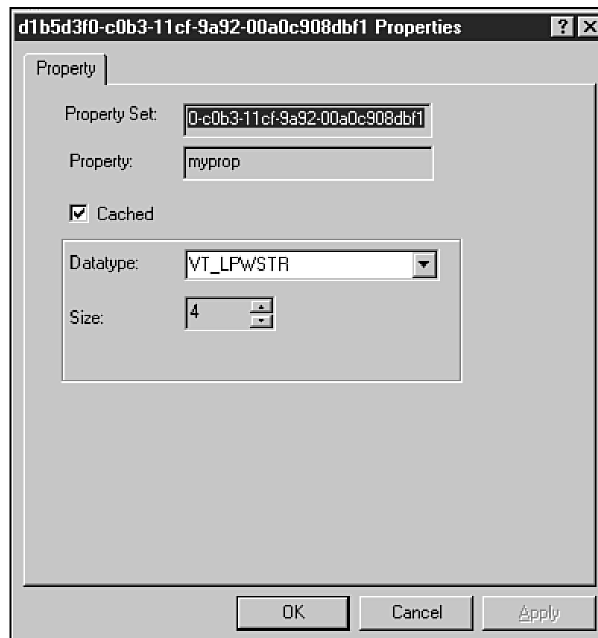
כדי להוסיף מאפיין מותאם אישית למסמך HTML, השתמש בתגית <META> של HTML. לדף הבא מאפיין מותאם אישית בשם myProp :

```
<html>
<head>
<META NAME="myProp" CONTENT="red">
<title>Some Document</title>
</head>
<body>
Hello!
</body>
</html>
```

התגית <META> מוספת בין תגיות <HEAD> של מסמך HTML. בדוגמה זו, התגית <META> משמשת להגדרת מאפיין בשם myProp עם הערך red. אתה יכול, כמובן, ליצור מאפיין עם כל שם וערך שתרצה.

לאחר הוספת מאפיין מותאם אישית לדף HTML, עליך לעצב את Index Server כדי שיזהה את המאפיין החדש. כדי לעשות זאת עליך להשתמש ב-Index Server Manager. בצע את השלבים הבאים :

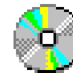
1. הפעל את Index Server Manager ובחר בתיקיה Properties.
2. המאפיין החדש צריך להופיע ברשימת המאפיינים במסגרת הימנית. אם המאפיין אינו מופיע, תוכל לחכות עד אשר Index Server יזהה אותו, או שתוכל להכריח את Index Server לבצע סריקה מחדש של הספריה המכילה את המסמך עם המאפיין החדש. כדי להכריח סריקה מחדש, בחר בשם הספריה ובאפשרות Action, Rescan (אולי תצטרך גם לסגור ולהפעיל מחדש את Index Server Manager).
3. בחר את המאפיין מרשימת המאפיינים ופתח את הכרטיסיה Properties של המאפיין. סמן את התיבה Cached ובחר בטיפוס הנתונים VT_LPWSTR עבור המאפיין (ראה תרשים 15.3).




**תרשים 15.3** הוספת מאפיין מותאם אישית

**4.** בחר בכרטיסיה Properties פעם נוספת, ובחר באפשרות Action, Commit changes.

סדרת פעולות זו מוסיפה מאפיין מותאם אישית חדש למטמון המאפיינים. לאחר שהוספת לראשונה מאפיין מותאם אישית, המאפיין מכיל את הערך NULL עבור כל המסמכים, עד אשר הם ימופתחו שוב.



לאחר ש-Index Server זיהה את המאפיין, ניתן להשתמש בו בשאילתות חיפוש בדיוק באותו אופן בו ניתן להשתמש במאפיינים סטנדרטיים, כגון, DocTitle ו-Size. התסריט בתדפיס 15.2 (נמצא בתקליטור תחת השם customProp.asp) מאחזר את כל מסמכי HTML אשר בהם ערך המאפיין myProp שווה ל-green.

עליך לשים לב לשורה המוזרה והלא מובנת הבאה אשר מופיעה בתסריט:

```
Con.Execute "SET PROPERTYNAME 'd1b5d3f0-c0b3-11cf-9a92-00a0c908dbf1'
↳ PROPID 'myProp' AS myProp"
```

מטרתה של הצהרה זו היא לקשר את המאפיין המותאם אישית לשם ידידותי שבו תוכל להשתמש בשאילתות שלך. מחרוזת התווים והספרות הארוכה מייצגת את ה-GUID (מזהה ייחודי גלובלי - Globally Unique Identifier) עבור מאפיינים המקושרים עם מסמכי HTML.

לדוגמה, אם אתה יוצר מאפיין מותאם אישית בשם favColor, עליך להשתמש בהצהרה הבאה כדי לספק למאפיין favColor שם ידידותי (אותו GUID משמש תמיד עבור מסמכי HTML):

```
Con.Execute "SET PROPERTYNAME 'd1b5d3f0-c0b3-11cf-9a92-00a0c908dbf1'
↳PROPID 'favColor' AS favColor"
```

**תדפיס 15.2** customProp.asp

```
<%
Set Con=Server.CreateObject( "adodb.Connection" )
Con.Open "provider=msidxs"
Con.Execute "SET PROPERTYNAME 'd1b5d3f0-c0b3-11cf-9a92-00a0c908dbf1'
  ↳ PROPID 'myProp' AS myProp"
mySQL = "SELECT  DocTitle, myProp " &
  ↳ "FROM  SCOPE() " &
  ↳ "WHERE FREETEXT(myProp, '" & "green" & "') > 0 "
Set RS = Con.Execute( mySQL )
WHILE NOT RS.EOF
%>
<p><%=RS( "DocTitle" )%>
<br><%=RS( "myProp" )%>
<%
RS.MoveNext
WEND
%>
```

# שימוש ב-ADO עם SQL 7.0 Full-Text Search

אם אתה יוצר אתר אינטרנט מונחה מסד נתונים באמצעות ASP, רוב הנתונים לא יישמרו בדפי HTML סטטיים אלא בטבלה אחת או במספר טבלאות מסד נתונים.

לדוגמה, דמיין לעצמך שאתה בונה אתר משרות עבור יועצים ל-ASP. קרוב לוודאי שתרצה לאפשר למבקרים לשלוח את קורות החיים שלהם דרך טופס מאובטח. השיטה ההגיונית והיעילה ביותר לאיחסון קורות חיים אלה היא באמצעות הצבתם בעמודת טקסט של טבלת מסד נתונים.

שיטה זו יוצרת בעיה כאשר משתמשים בשרתי חיפוש מסורתיים כמו Microsoft Index Server. כיון שניתן להשתמש ב-Microsoft Index Server רק כדי למפתח (To Index) מסמכים סטטיים, לא תוכל להשתמש בו למיפתוח קורות החיים אשר נשלחו לאתר שלך. Microsoft Index Server ייכשל במיפתוח רוב הנתונים החשובים.

למרבה המזל, Microsoft זיהתה בעיה זו. כדי לפתור את הבעיה למיפתוח נתונים המאוחסנים בטבלאות מסד נתונים, הציגה Microsoft את השירות **חיפוש טקסט-מלא - Full-Text Search**, עם Microsoft SQL Server 7.0. השירות Full-Text Search יכול לשמש למיפתוח ולביצוע שאילתות מול עמודות טקסט בטבלת מסד נתונים כלשהי של SQL 7.0.

# לשם מה נדרש חיפוש טקסט-מלא (Full-Text Search)?

ודאי תתמה האם שרות Full-Text Search באמת נחוץ. Transact-SQL, השפה של SQL Server, כבר תומכת בהתאמת תווים באמצעות השימוש בתווים כלליים (Wildcard Characters) ובאופרטור LIKE. לדוגמה, תוכל להשתמש בהצהרה הבאה של SQL כדי לאחזר את הטקסט של כל קורות החיים המכילים את הביטוי "will work cheap" מטבלה בשם Consultants:

```
SELECT resume FROM Consultants
WHERE resume LIKE '%will work cheap%'
```

הצהרה זו פועלת באופן מושלם. היא משתמשת בתו אחוז (%) כדי להתאים מחרוזת בכל גודל שהוא המקדימה או הבאה אחרי הביטוי. השאילתה תחזיר את קורות החיים המכילים התאמה מושלמת לביטוי זה.

ישנם שני חסרונות לשימוש בגישה זו לחיפוש בתוכנה של טבלת מסד נתונים. הראשון, והחשוב ביותר, השאילתה אינה יעילה במיוחד. אם יש לך טבלה המכילה אלפי (או עשרות אלפי) קורות חיים, שימוש באופרטור LIKE באופן זה יגרום למסד הנתונים שלך לזחול. SQL Server פשוט לא תוכנן לטפל ביעילות בסוג של שאילתה כזו.

דבר נוסף, שימוש באופרטור LIKE יגביל אותך לביצוע חיפוש רק כדי לאחזר התאמות מדויקות. לדוגמה, השאילתה LIKE שבדוגמה הקודמת לא תמצא התאמה לקורות חיים המכילים את הביטוי "will work very cheap" או "will work very, very cheap".

משתמשים מצפים מאתרי אינטרנט מסחריים לתמוך ביכולות חיפוש מתקדמות כמו למשל בחיפוש בוליאני ובהתאמת תבניות מעורפלות (Fuzzy Patterns). משתמשים מצפים להיות מסוגלים לבצע את אותם סוגי חיפוש אשר הם יכולים לבצע ב-!Yahoo או ב-AltaVista. לא ניתן לבצע סוגי שאילתות אלה באמצעות Transact-SQL (או שהם יהיו כל כך מורכבים ולא יעילים כך שלא ישתלם לבצעם).

חיפוש טקסט-מלא - Microsoft Full-Text Search, אינו שותף לליקויים אלה. הוא משתמש באינדקסים משלו אשר תוכננו במיוחד לטיפול בחיפושים מורכבים. חיפוש טקסט-מלא תומך הן בתבניות מעורפלות והן בשאילתות בוליאניות.

# התקנה ואפשור חיפוש טקסט-מלא (Full-Text Search)

השירות חיפוש טקסט-מלא נכלל עם Microsoft SQL 7.0. למרות שהשירות נכלל עם כל הגרסאות של SQL 7.0, הוא לא יפעל עם הגירסה Desktop. לא ניתן להשתמש בחיפוש טקסט-מלא עם Windows 95/98 או עם Windows NT Workstation. השירות יפעל אך ורק עם Windows NT Server.

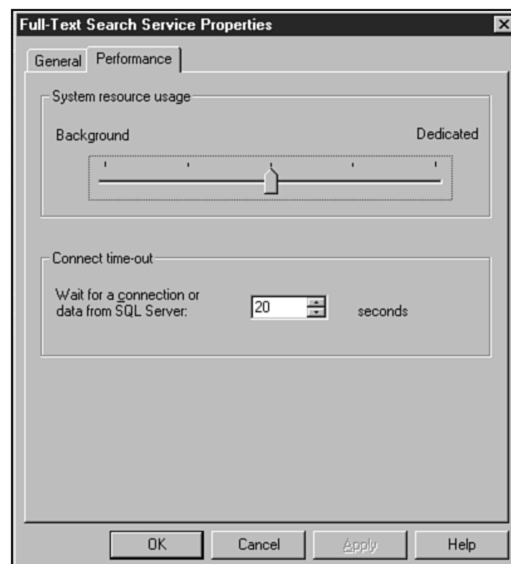
חיפוש טקסט-מלא אינו מותקן כברירת מחדל בעת התקנת SQL Server. במהלך ההתקנה, עליך לבחור לבצע התקנה מותאמת (Custom Installation) ולבחור

ב-Full-Text Search Service מתפריט הרכיבים הניתנים להתקנה. לחילופין, ניתן להתקין חיפוש טקסט-מלא לאחר התקנת SQL Server באמצעות הפעלה חוזרת של תוכנת ההתקנה.

לאחר התקנת השירות, הוא יירשם ב-SQL Server Service Manager תחת השם Microsoft Search. לפני שתוכל להשתמש בשירות, עליך לאפשר אותו באמצעות לחיצה על Start/Continue יחד עם בחירה בשירות (ראה תרשים 15.4).

**תרשים 15.4** הפעלת שירות חיפוש טקסט-מלא

כמו כן, באפשרותך לציין את כמות משאבי המערכת אשר בהם יוכל חיפוש טקסט-מלא להשתמש. כדי לקבוע אפשרות זו, הפעל את SQL Server Enterprise Manager, לחץ על Support Services, ולחץ לחיצה ימנית על Full-Text Search כדי להציג את דף המאפיינים של השירות Full-Text Search. בדף המאפיינים, בחר בכרטיסיה Performance וספק ערך עבור System Resource Usage (ראה תרשים 15.5).




**תרשים 15.5** קביעת השימוש במשאבי החיפוש

# הוספת אינדקס טקסט-מלא לטבלה (Full-Text Search)

ניתן להגדיר וליצור אינדקס טקסט-מלא (Full-Text Index) עבור טבלה באמצעות שימוש ב-SQL Server Enterprise Manager או באמצעות הפעלת שגרות מערכת מאוחסנות. השימוש ב-SQL Server Enterprise Manager ליצירת אינדקס הוא קל הרבה יותר, ולכן זוהי השיטה שתתואר כאן.

לפני הוספת תמיכה לחיפוש טקסט-מלא עבור טבלה, עליך להיות מודע לעובדה שלא תוכל לשנות את הטבלה לאחר הוספת האינדקס. לדוגמה, לא תוכל לשנות אף אחד מהאינדקסים הנורמליים של הטבלה או לשנות את עמודות הטבלה (תוכל בכל מקרה לשנות את הנתונים בטבלה). כדי לבצע שינויים לטבלה בעלת אינדקס טקסט-מלא, עליך להסיר את האינדקס, לבצע את השינויים, ולהוסיף את האינדקס שוב. זה מקשה על החיים במקרה של אתר אינטרנט פעיל, כיון שהמשמעות היא שלא ניתן לבצע שינויים בטבלה ללא נטרול זמני של דף החיפוש של האתר.

כמו כן, עליך להיות מודע לכך שלא ניתן להוסיף אינדקס טקסט-מלא לטבלה אלא אם כן יש לה אינדקס ייחודי, בעל עמודה בודדת. כדי ליצור אינדקס כזה עבור טבלה, ניתן להוסיף לה מפתח ראשי (Primary Key). בדרך כלל מגדירים מפתח ראשי עבור עמודת זיהוי (Identity Column) כדי להבטיח את ייחודיותם של כל ערכי העמודה.

כדי ליצור אינדקס טקסט-מלא עבור טבלה, הפעל את Microsoft Enterprise Manager ובחר בטבלה. כעת, לחץ לחיצה ימנית על הטבלה ובחר באפשרות Full-Text Index Table, Define Full-Text Indexing on a Table. סדרת פעולות אלו יפעילו את האשף Full-Text Indexing Wizard (ראה תרשים 15.6). האשף ינחה אותך במהלך השלבים הבאים :

1. **בחירת אינדקס.** כל טבלה בעלת אינדקס טקסט-מלא חייבת לכלול עמודה בעלת ערכים ייחודיים. עמודה זו נקראת עמודת המפתח (Key column). כאשר אתה מבצע שאילתת טקסט-מלא, ניתן להחזיר את ערכי עמודת המפתח עבור השורות המתאימות לשאילתה.

**תרשים 15.6** יצירת אינדקס טקסט-מלא (Full-Text Index)

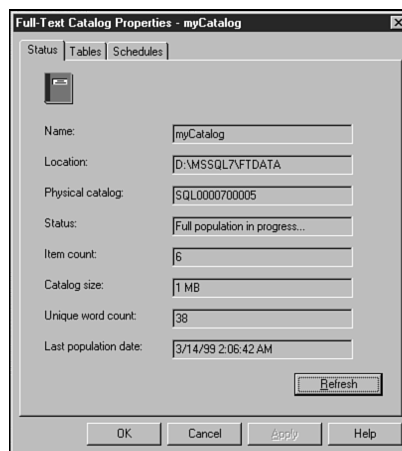
**.2** בחירת עמודות טבלה. ניתן ליצור אינדקס טקסט-מלא עבור מספר עמודות בטבלה. לדוגמה, ניתן ליצור אינדקס עבור עמודות VARCHAR ,CHAR ו-TEXT.

**.3** בחירת קטלוג מפורט. אינדקס טקסט-מלא מאוחסן בקטלוג. באפשרותך ליצור קטלוג חדש או לבחור בקטלוג קיים. למרות שניתן להציב אינדקסים רבים של טקסט-מלא באותו קטלוג, אל תעשה זאת כאשר אתה ממפתח טבלאות גדולות.

**.4** בחירת או יצירת **תזמוני אכלוס** (Population Schedules). שלא כמו אינדקס טבלה נורמלי, אינדקס טקסט-מלא אינו מעודכן באופן אוטומטי על ידי SQL Server. אם שורות חדשות נוספו לטבלה או שרשומות בטבלה שונו, יש לעדכן את אינדקס הטקסט המלא. כדי לעדכן אינדקס טקסט-מלא באופן מתוזמן, צור תזמוני אכלוס. תוכל לבחור לעדכן את האינדקס כל פעם מחדש (אכלוס מלא - Full population), או שתוכל לבחור לעדכן אינדקס רק עם שינויים חדשים (אכלוס מצטבר - Incremental population). בדרך כלל עליך לבחור באכלוס מצטבר.

לאחר שהשלמת את האשף לחץ Finish, ואינדקס הטקסט המלא החדש ייווצר בקטלוג שציינת. אינדקס הנוצר לראשונה הוא ריק. לפני שתוכל להשתמש בו, עליך לאכלס אותו. כדי לאכלס אינדקס, בחר ב-Full Text Catalog, לחץ לחיצה ימנית על שמו של הקטלוג, ובחר ב-Start Population. פעולה זו תתחיל את תהליך האכלוס.

עבור טבלאות גדולות, אכלוס האינדקס בפעם הראשונה עלול להימשך זמן רב. תוכל לצפות בתהליך בזמן אמת באמצעות לחיצה כפולה על שם הקטלוג וצפייה במונה הפריט. בכל פעם שממופתחת שורה חדשה בטבלה, יגדל מונה הפריט באחד (ראה תרשים 15.7).




**תרשים 15.7** צפייה במונה הפריט

# ביצוע שאילתות טקסט-מלא

כדי לתמוך בשאילתות טקסט-מלא, נוספו ל-Transact-SQL שתי מילות מפתח חדשות ושתי פונקציות חדשות. באמצעות שימוש במילת המפתח CONTAINS ובפונקציה CONTAINSTABLE, תוכל לבצע שאילתות בוליאניות וסוגים נוספים של שאילתות חיפוש מדויקות מאוד. באמצעות שימוש במילת המפתח FREETEXT או בפונקציה FREETEXTTABLE, תוכל לבצע חיפושים במבנה חופשי.

## CONTAINS Predicate

מילת המפתח CONTAINS גמישה מאוד. תוכל להשתמש במילת מפתח זו לביצוע כל אחד מסוגי החיפוש הבאים:

- **חיפוש בוליאני** (Boolean Search) - בחיפוש בוליאני, ניתן להשתמש במילים AND, OR, ו-AND NOT בביטוי החיפוש. לדוגמה, תוכל לבצע חיפוש בטבלת קורות חיים אחר כל קורות החיים המכילים את הביטוי Tel-Aviv או Jerusalem.
- **חיפוש ביטוי מדויק** (Exact Phrase Search) - באמצעות הצבת מרכאות (" ") מסביב לביטוי החיפוש, ניתן להחזיר רק את הרשומות המתאימות בדיוק לביטוי. לדוגמה, תוכל לחפש דפי ASP ולא להחזיר רשומות המכילות את הביטוי old book pages.
- **חיפוש משוקלל** (Weighted Search) - בחיפוש משוקלל תוכל לקבוע משקל שונה עבור הביטויים המופיעים בביטוי החיפוש. לדוגמה, בחיפוש אחר קורות חיים המכילים את המילים ASP ו-PERL, תוכל לתת משקל רב יותר למילה ASP מאשר למילה PERL.
- **חיפוש נטייתי** (Inflectional Search) - בחיפוש נטייתי, יכולה להתקיים התאמה גם במקרה של שוני דקדוקי במילת החיפוש. שוני זה כולל הן צורת רבים וצורת יחיד של שמות עצם, והן זמנים דקדוקיים שונים של פעלים. חיפוש של המילה house יחזיר גם שורות המכילות את המילה houses. חיפוש של המילה open יחזיר גם שורות המכילות את המילה opened.
- **חיפוש קירבה** (Proximity Search) - בחיפוש קירבה, ניתן להעניק דרגה גבוהה יותר להתאמות בהן המילים בביטוי החיפוש מופיעות בקירבה רבה יותר. לדוגמה, תוכל לתת ערך גבוה יותר לקורות חיים בהם המילים very ו-cheap קרובות אחת לשנייה מאשר לקורות חיים בהן מילים אלו רחוקות האחת מהשנייה.

לדוגמה, נניח שעליך לבצע שאילתה אשר מחזירה מטבלת קורות החיים את כל הרשומות המכילות את המילים Tel-Aviv או Jerusalem. תוכל לבצע את החיפוש עם מילת המפתח CONTAINS בעזרת ההצהרה הבאה:

```
SELECT c_resume FROM Consultants WHERE
CONTAINS( c_resume, '"Tel-Aviv" OR "Jerusalem"' )
```

הצהרה זו מחזירה את כל קורות החיים מהטבלה Consultants כאשר העמודה c_resume מכילה את המילים Tel-Aviv או Jerusalem. מילת המפתח CONTAINS מקבלת שני ארגומנטים. הארגומנט הראשון מספק את רשימת העמודות בהן יתבצע החיפוש (ניתן להשתמש ב-* עבור כל העמודות). הארגומנט השני מספק את ביטוי החיפוש.

להלן כיצד קריטריון החיפוש בדוגמה הקודמת יכול לשמש בתסריט ASP :

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" —>
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
mySQL = "SELECT c_resume FROM Consultants WHERE " &
  ➥ "CONTAINS( c_resume, '""San Francisco"" OR ""Modesto""' )"
Set RS = Con.Execute( mySQL )
While NOT RS.EOF
  Response.Write RS( "c_resume" ) & "<br>"
  RS.MoveNext
WEND
%>
```

שים לב לדברים המשונים אשר חייבים להיעשות עם סימני המרכאות כדי שדוגמה זו אכן תפעל. VBScript משתמש בסימני מרכאות לסימון התחלה וסיומת של מחרוזת. אולם, אם תשתמש בשתי מרכאות בשורה, VBScript יפרש את המרכאות כחלק מהמחרוזת.

## שימוש בפונקציה CONTAINSTABLE

באמצעות הפונקציה CONTAINSTABLE ניתן לבצע את אותם סוגי חיפוש שביצעת באמצעות מילת המפתח CONTAINS. בכל מקרה, הפונקציה CONTAINSTABLE מחזירה טבלה במקום סדרת שורות תואמות.

יתרון השימוש בפונקציה CONTAINSTABLE הוא שניתן להשתמש בפונקציה זו כדי לדרג את רשימת התוצאות. הטבלה המוחזרת על ידי פונקציה זו מכילה שתי עמודות הנקראות Key ו-Rank. העמודה Key מכילה ערך ייחודי עבור כל שורה, והעמודה Rank מכילה את הדירוג השייך לכל שורה. כל שורה מדורגת בטווח הערכים 0-1000 (הדרגה הגבוהה יותר טובה יותר).

לדוגמה, הטבלה הבאה מאחזרת מהטבלה Consultants את כל קורות החיים המכילים את המילה Active או Server, ומדרגת את התוצאות :

```
SELECT Rank, c_resume FROM
Consultants
INNER JOIN
CONTAINSTABLE( Consultants, c_resume, 'Active or Server' )
AS SR
ON c_id = [Key]
ORDER BY Rank DESC
```

הצהרה זו די מורכבת והיא דורשת כמה הסברים. כאשר טבלה מוחזרת על ידי הפונקציה CONTAINSTABLE, היא מכילה אך ורק את העמודה Key ואת העמודה Rank. אם ברצונך להציג את הרשומות אשר התאימו לחיפוש, עליך לאחד טבלה זו עם הטבלה עליה התבצע החיפוש. בדוגמה הקודמת, הטבלה Consultants מאוחדת עם

הטבלה אשר הוחזרה על ידי הפונקציה CONTAINSTABLE באמצעות שימוש בעמודה c_id של הטבלה Consultants והעמודה Key בטבלה CONTAINSTABLE.

בדוגמה הקודמת, העמודה Rank משמשת לסידור מדורג של תוצאות החיפוש אשר הוחזרו על ידי השאילתה. לדוגמה, אם קורות חיים מסוימים מכילים גם את המילה Active וגם את המילה Server, הם ידורגו גבוה יותר מאשר קורות חיים המכילים רק את המילה Server. דירוגן של השורות נשלח כפלט עם התוכן הממשי של קורות החיים.

הפונקציה CONTAINSTABLE מקבלת שלושה ארגומנטים. הארגומנט הראשון מכיל את שם הטבלה עליה יתבצע החיפוש. הארגומנט השני מכיל רשימה של כל העמודות בהן יתבצע החיפוש (ניתן להשתמש ב-* עבור כל העמודות). לבסוף, הארגומנט השלישי מכיל את ביטוי החיפוש עצמו.

## FREETEXT Predicate

מרבית האנשים אינם משתמשים בחיפוש בקנה מידה מתוחכם כאשר הם מבצעים חיפוש בדף אינטרנט. לעיתים רחוקות אנשים מבצעים חיפושים בוליאניים או חיפושי קירבה. בדרך כלל, הם פשוט מקלידים אוסף מילים ומקווים להתאמות טובות.

מילת המפתח FREETEXT שימושית מאוד לביצוע חיפושים במבנה חופשי (Free-Form searches). מילת מפתח זו תוכננה לביצוע התאמה של משמעות המילים בביטוי החיפוש יותר מאשר התאמה מדויקת שלהם. לפניך דוגמה לשימוש במילת המפתח FREETEXT לחיפוש אחר קורות החיים בעלי התאמה ל-Active Server Pages :

```
SELECT c_resume
FROM Consultants WHERE
FREETEXT( c_resume, 'Active Server Pages' )
```

תוכל להקליד כל דבר כביטוי החיפוש מבלי לקבל הודעת שגיאה. מילת המפתח FREETEXT תנסה תמיד לקבל התאמות טובות.

בכל אופן, אם תקליד ביטוי חיפוש בוליאני, מילת המפתח תתעלם מהאופרטורים OR, AND ו-AND NOT. לא ניתן להשתמש במילת המפתח FREETEXT כדי לבצע את אותם סוגי חיפוש שניתן לבצע באמצעות מילת המפתח CONTAINS.

## שימוש בפונקציה FREETEXTTABLE

באמצעות הפונקציה FREETEXTTABLE ניתן לבצע את אותם סוגי חיפוש שביצעת באמצעות מילת המפתח FREETEXT. בכל אופן, הפונקציה FREETEXTTABLE מחזירה טבלה במקום סדרה של רשומות תואמות.

הטבלה המוחזרת על ידי פונקציה זו מכילה שתי עמודות בשם Key ו-Rank. העמודה Key מזהה כל שורה בתוצאות החיפוש באופן ייחודי. העמודה Rank מספקת דירוג יחסי לכל תוצאה. להלן דוגמה לשימוש בפונקציה זו :

```
SELECT Rank, c_resume FROM
Consultants
INNER JOIN
FREETEXTTABLE( Consultants, c_resume,
  'Active Server Pages' )
AS SR
ON c_id = [Key]
ORDER BY Rank DESC
```

הצהרה זו מאחזרת את כל קורות החיים אשר מתאימים לקריטריון החיפוש Active Server Pages. הטבלה המוחזרת מהפונקציה FREETEXTTABLE מאוחדת עם הטבלה Consultants באמצעות שימוש בעמודות c_id ו-Key.

הפונקציה FREETEXTTABLE מקבלת שלושה ארגומנטים. הארגומנט הראשון מכיל את שם הטבלה עליה יתבצע החיפוש. הארגומנט השני מכיל רשימה של כל העמודות בהן יתבצע החיפוש (ניתן להשתמש ב-* עבור כל העמודות). הארגומנט השלישי מכיל את ביטוי החיפוש עצמו.

להלן דוגמה לאופן בו ניתן ליישם שאילתה זו בתסריט ASP:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
mySQL = "SELECT Rank, c_resume FROM " &
 ↳ "Consultants " &
 ↳ "INNER JOIN " &
 ↳ "FREETEXTTABLE( Consultants, c_resume, " &
 ↳ " 'Active Server Pages' )" &
 ↳ "AS SR " &
 ↳ "ON c_id = [Key] " &
 ↳ "ORDER BY Rank DESC "
Set RS = Con.Execute( mySQL )
While NOT RS.EOF
 Response.Write RS( "c_resume" ) & "<br>"
 RS.MoveNext
WEND
%>
```

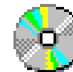
# אחזור סטטיסטיקות אינדקס

אם תלחץ לחיצה כפולה על שמו של קטלוג החיפוש שלך ב-SQL Server Enterprise Manager, תוכל לראות מספר נתונים סטטיסטיים על מצבו הנוכחי של אינדקס הטקסט המלא. לדוגמה, תוכל לראות את מונה הפריט (מספר שורות הטבלה שמופתחו עד כה), מונה האינדקס הייחודי (מספר המילים הייחודיות שמופתחו) ואת מצב האכלוס (האם כרגע מתבצע אכלוס מלא או אכלוס מצטבר).

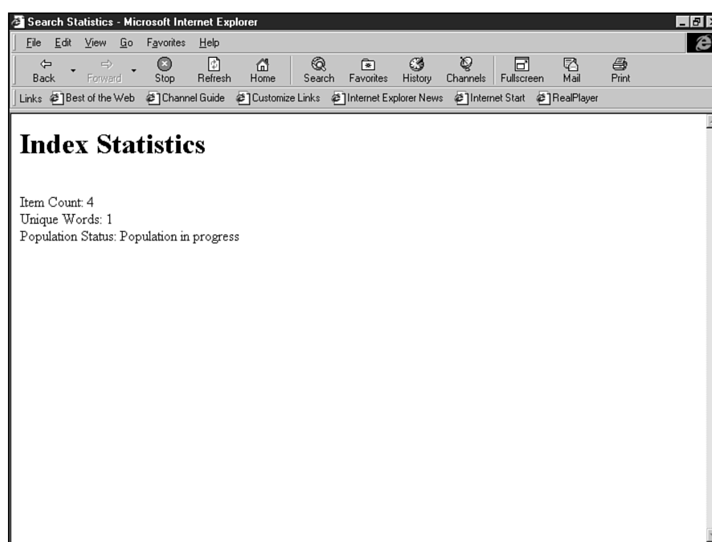
תוכל לאחזר ערכים דומים מאוד אלה באמצעות שימוש בשגרת המערכת המאוחסנת FULLTEXTCATALOGPROPERTY. ההצהרה הבאה, לדוגמה, מחזירה את מספר הפריטים שמופתחו עד כה:

```
SELECT FULLTEXTCATALOGPROPERTY('myCatalog', 'ItemCount')
```

החלף את המילה myCatalog בשמו של קטלוג הטקסט המלא שלך. כאשר תבצע הצהרה זו יוחזר מספר השורות שמופתחו עד כה.



ניתן להשתמש בשגרה FULLTEXTCATALOGPROPERTY ליצירת דף סטטיסטיקות עבור אתר האינטרנט שלך המציג את מצבו הנוכחי של אינדקס הטקסט המלא. התסריט בתדפיס 15.3 (נמצא בתקליטור תחת השם searchState.asp) מציג בדפדפן מספר נתונים סטטיסטיים זמינים (ראה תרשים 15.8).




**תרשים 15.8** אחזור סטטיסטיקות חיפוש

**תדפיס 15.3** searchState.asp

```
<html>
<head><title>Search Statistics</title></head>
<body>
<%
Function showPop( thePopStatus )
  SELECT CASE thePopStatus
    CASE "0"
    showPop = "Idle"
    CASE "1"
showPop = "Population in progress"
    CASE "2"
    showPop = "Paused"
    CASE "3"
    showPop = "Throttled"
```

```
    CASE "4"
    showPop = "Recovering"
    CASE "5"
    showPop = "Shutdown"
    CASE "6"
    showPop = "Incremental population"
    CASE "7"
    showPop = "Updating Index"
  END SELECT
End Function

Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=TEST"
mySQL = "SELECT FULLTEXTCATALOGPROPERTY('myCatalog', 'ItemCount') ICount"
Set RS1 = Con.Execute( mySQL )
mySQL = "SELECT FULLTEXTCATALOGPROPERTY('myCatalog',
  ↳'UniqueKeyCount') UKey"
Set RS2 = Con.Execute( mySQL )
mySQL = "SELECT FULLTEXTCATALOGPROPERTY('myCatalog',
  ↳'PopulateStatus') PStatus"
Set RS3 = Con.Execute( mySQL )
%>
<h1>Index Statistics</h1>
<br>Item Count: <%=RS1( "ICount" )%>
<br>Unique Words: <%=RS2( "UKey" )%>
<br>Population Status: <%=showPop( RS3( "PStatus" ) )%>

</body>
</html>
```

## שימוש בחיפוש טקסט-מלא ליצירת דף חיפוש

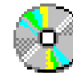
סעיף זה מציג דוגמה המראה כיצד תוכל ליצור דף חיפוש פשוט באמצעות שימוש בחיפוש טקסט-מלא. דף חיפוש זה מאפשר לך לחפש קורות חיים עבור מועמדים לעבודה המתאימים לקריטריון החיפוש שלך. דף החיפוש תומך הן בחיפוש בוליאני והן בחיפוש במבנה חופשי.

שמה של הטבלה בדוגמה זו היא Consultants. ניתן ליצור אותה באמצעות הצהרת SQL הבאה:

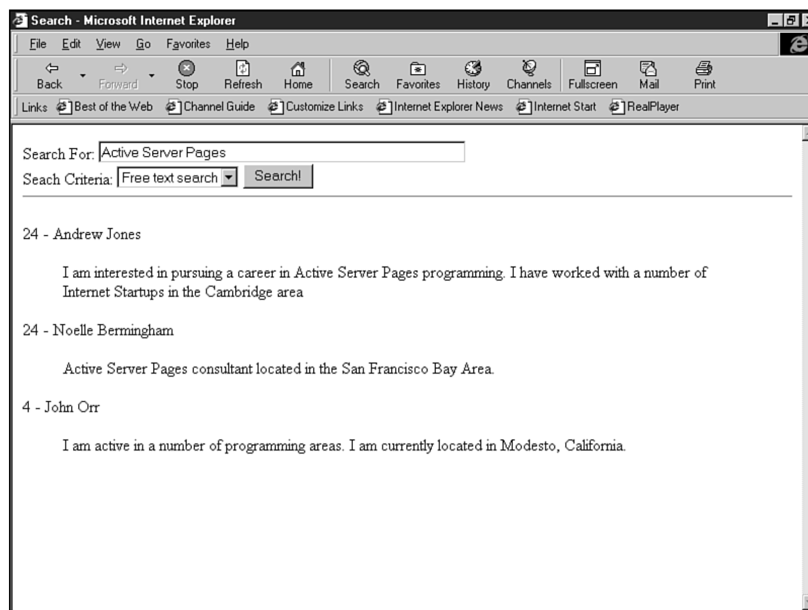
```
CREATE TABLE Consultants (
  c_id INT IDENTITY PRIMARY KEY,
  c_name VARCHAR( 50 ),
  c_resume TEXT
)
```

טבלה זו מכילה שלוש עמודות. העמודה הראשונה היא עמודת זיהוי, המשמשת לזיהוי ייחודי של כל שורה בטבלה (חובה ליצור עמודה ייחודית כדי להשתמש בחיפוש טקסט-מלא). העמודה השנייה מכילה את שמות האנשים אשר שלחו את קורות החיים. לבסוף, העמודה השלישית מכילה את הטקסט הממשי של קורות החיים.

לאחר יצירת טבלה זו, צור עבורה אינדקס טקסט-מלא, הוסף מספר קורות חיים ואכלס את האינדקס. ראה את שני הסעיפים, **התקנה ואפשור חיפוש טקסט-מלא (Full-Text Search)** ו-**הוספת אינדקס טקסט-מלא לטבלה**, מוקדם יותר בפרק זה, כדי ללמוד כיצד לבצע זאת.



הקוד המלא של דף החיפוש מופיע בתדפיס 15.4 (נמצא בתקליטור תחת השם search.asp). דף החיפוש כולל טופס המכיל שדה קלט טקסט ורשימת בחירה. רשימת הבחירה משמשת לבחירת סוג החיפוש: בוליאני או מבנה חופשי. תוצאות החיפוש מוצגות מתחת לטופס (ראה תרשים 15.9).




**תרשים 15.9** דף חיפוש טקסט-מלא

שים לב לדבר אחד מסוים בקוד זה. כאשר ביטוי החיפוש מאוחזר מטופס ה-HTML, מוסרים ממנו כל סימני הגרש. אם תכלול סימני גרש בביטוי החיפוש, הדבר יגרום ליצירת הודעת שגיאה כאשר הביטוי יישלח ל-SQL Server.

כמו כן, עליך להיות מודע לעובדה שדף החיפוש יגרום ליצירת שגיאה אם תנסה לבצע חיפוש בוליאני ללא שימוש בביטוי חיפוש בוליאני. כדי לעקוף בעיה זו עליך ללכוד שגיאות באמצעות שימוש בהצהרה ON ERROR RESUME next ובאובייקט ERR. כאשר תבצע חיפוש במבנה חופשי, בכל אופן, תוכל להקליד כל ביטוי שתרצה מבלי לקבל שגיאה.

**תדפיס 15.4** search.asp

```
<html>
<head><title> Search </title></head>
<body>
<form method="post" action="search.asp">
Search For:
<input name="search" type="text" size=50>
<br>Search Criteria:
<select name="criteria">
<option value="0">Free text search
<option value="1">Boolean search
</select>
<input type="submit" value="Search!">
<hr>
<%
search = TRIM( Request( "search" ) )
criteria = TRIM( Request( "criteria" ) )
if search <> "" then
  search = Replace( search, "'", "" )
  Set Con = Server.CreateObject( "ADODB.Connection" )
  Con.Open "FILE NAME=c:\myDataLink.UDL"
  mySQL = "SELECT Rank, c_name, c_resume FROM " &
   ↳ "Consultants INNER JOIN "
  if criteria = "0" then
    mySQL = mySQL & "FREETEXTTABLE( Consultants, *, "
  else
    mySQL = mySQL & "CONTAINSTABLE( Consultants, *, "
  end if
  mySQL = mySQL & "'" & search & "' )" &
    ↳ "AS SR " &
    ↳ "ON c_id = [Key] " &
    ↳ "ORDER BY Rank DESC "
Set RS = Con.Execute( mySQL )
if RS.EOF then
  Response.Write "No matches found!"
else
  While NOT RS.EOF
  %>
  <p><%=RS( "Rank" ) %> - <%=RS( "c_name" )%>
  <blockquote><%=RS( "c_resume" )%></blockquote>
  <%
```

```
  RS.MoveNext
 WEND
end if
end if
%>
```

# סיכום

בפרק זה למדת שתי שיטות להוספת דף חיפוש לאתר האינטרנט שלך. בחלק הראשון, למדת כיצד להשתמש ב-Microsoft Index Server כדי למפתח את המסמכים הסטטיים בשרת האינטרנט שלך. כמו כן למדת כיצד להשתמש ב-Index Server עם ADO כדי לבצע חיפושים. בחלק השני, למדת כיצד להשתמש בחיפוש טקסט-מלא (Full-Text Search) כדי למפתח ולחפש נתונים אשר מאוחסנים בטבלת מסד נתונים.

# חלק 4

# רכיבים מותאמים של ASP

**בחלק זה:**

- **רכיבים של צד שלישי (Third-Party Components)**
- **יצירת Windows Script Components**
- **יצירת רכיבים באמצעות Visual Basic**
- **יצירת רכיבים באמצעות Visual C++**

## פרק 16

# רכיבים של צד שלישי

**בפרק זה:**

- רכיבים גרפיים
- רכיבים להעברת קבצים
- רכיבי רשת
- רכיבים נוספים

כמפתח דפי ASP, אין דבר מלהיב יותר מאשר לגלות רכיבים חדשים של ASP. רכיבים של צד שלישי (Third-Part Components), שפותחו על ידי מי שאינם Microsoft) פותחים אפשרויות תכנות חדשות על ידי הרחבת הערכה הבסיסית של הרכיבים הפנימיים הניתנים להתקנה, אשר נכללים בשרת אינטרנט – IIS - Internet Information Server.

האם ברצונך לבקש דף מאתר אינטרנט אחר? אין בעיה. כמה רכיבים של צד שלישי מאפשרים לך למשוך דף מאתר אינטרנט אחר על ידי שימוש בפרוטוקול HTTP. האם ברצונך ליצור תמונה מותאמת בדף ASP שלך? שוב, ישנם רכיבים המאפשרים לך לצייר בצורה דינמית תמונות מכל סוג בדף ASP. אם ברצונך לבצע דבר מה בדף ASP שלך אשר אינו נתמך על ידי אחד הרכיבים הבסיסיים של ASP, מירב הסיכויים הם שרכיב כזה כבר פותח על ידי מישהו כדי לתמוך בדרישה זו.

פרק זה מתמקד בשלושה רכיבים מועילים במיוחד של ASP. בחלק הראשון של הפרק, **רכיבים גרפיים**, יוצג הרכיב הגרפי ShotGraph. רכיב זה מאפשר לך ליצור תמונות JPEG מותאמות על המקום. ביכולתך להשתמש ברכיב זה ליצירת גרפים, כתגובה לשאילתות של המשתמש, או ליצור בצורה דינמית תמונות בעלות גופנים שונים.

בחלקו השני של הפרק, **רכיבים להעברת קבצים**, תלמד כיצד להשתמש בתוכנה Artisans SA-FileUp כדי לטפל בקלות בהעברת קובץ לאתר האינטרנט שלך. רכיב זה מאפשר לך לשמור קובץ שהועבר אליך, לקובץ או למסד נתונים, או לטפל בתכולת הקובץ בתוך דף ASP שלך.

בחלקו השלישי של הפרק, **רכיבי רשת**, תלמד כיצד להשתמש ברכיבים של WinSock כדי לעבוד עם הפרוטוקולים HTTP ו-NNTP. שימוש ברכיבים של WinSock מאפשרים להעביר טפסים או לאחזר דפי אינטרנט מאתר אינטרנט אחר. בנוסף, יכול רכיב WinSock להעביר או לאחזר מאמרים מקבוצות דיון, באמצעות פרוטוקול NNTP.

בחלקו הרביעי של פרק זה, **רכיבים שונים**, מופיעה סקירה קצרה של כמה רכיבים מעניינים של צד שלישי, אשר יכולים לעזור לך בעיצוב דף ASP שלך. פרק זה לא רק יציג לך סקירה של רכיבים אלה, אלא גם יספק לך רשימה של מקורות אפשריים באינטרנט שבהם תוכל למצוא רכיבים נוספים שיענו על דרישותיך.

# רכיבים גרפיים

התקליטור המצורף לספר זה מכיל גרסת התנסות של הרכיב ShotGraph. רכיב זה מכיל אוסף שיטות ומאפיינים לטיפול בתמונות JPEG. שימוש ברכיב ShotGraph מאפשר:

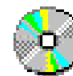
- יצירה תמונות דינמיות של ספרות, לשימוש במונה כניסות לאתר שלך.
- יצירת תרשים עוגה וגרפים נוספים, כתגובה לשאילתות של המשתמש.
- יצירה אוטומטית של תמונות ממוזערות (Thumbnails) לאוסף תמונות.
- הצגת טקסט בכל צורת גופן הזמינה בשרת האינטרנט שלך.
- שינוי גודל, סיבוב ושיקוף תמונות כתגובה לפעולות המשתמש.

ביסודו של דבר, מאפשר הרכיב ShotGraph לבצע על תמונות JPEG כמעט כל דבר העולה על רוחך. בחלק זה נידונות שתי דוגמאות לתוכניות המשתמשות ברכיב זה: בראשונה תלמד כיצד להשתמש ברכיב ShotGraph כדי ליצור מונה כניסות לאתר שלך, ובשנייה כיצד ליצור גרפים מטבלה של מסד נתונים.

**הערה:**

גרסת ההתנסות של הרכיב ShotGraph המצורפת לתקליטור זה מוגבלת לתמונות בגודל 320x240 פיקסלים. כמו כן מנוטרלות כמה פונקציות ותכונות של הרכיב.

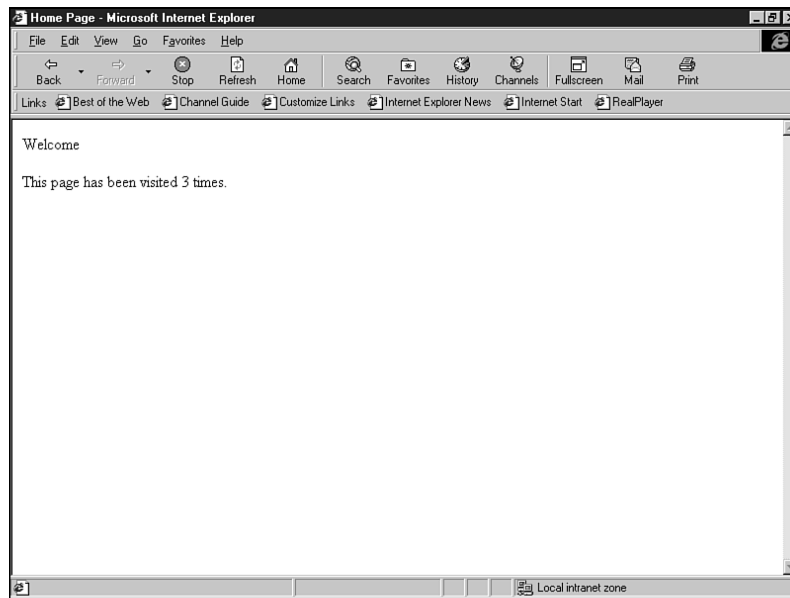
# יצירת מונה כניסות עם הרכיב ShotGraph



נניח שברצונך לעקוב ולהציג את מספר הפעמים שמישהו נכנס לאתר האינטרנט שלך. באפשרותך להציג מונה כניסות פשוט על ידי שימוש בתסריט ASP הבא (נמצא בתקליטור בשם plainHitCounter.asp):

```
<%
Application.Lock
Application( "home_hits" ) = Application( "home_hits" ) + 1
Application.UnLock
%>
<HTML>
<HEAD><TITLE> Home Page </TITLE></HEAD>
<BODY>
Welcome
<P>
This page has been visited <%=Application( "home_hits" )%> times.
</BODY>
</HTML>
```

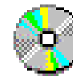
תסריט זה משתמש במשתנה יישום (Application Variable) כדי לאחסן את מספר הפעמים שבהם בוצעה גישה לעמוד האינטרנט. ערך המשתנה מוצג כפלט על ידי טקסט פשוט. כפי שהינך יכול לראות בתרשים 16.1, זו אינה הדרך המלהיבה ביותר להציג מידע זה.



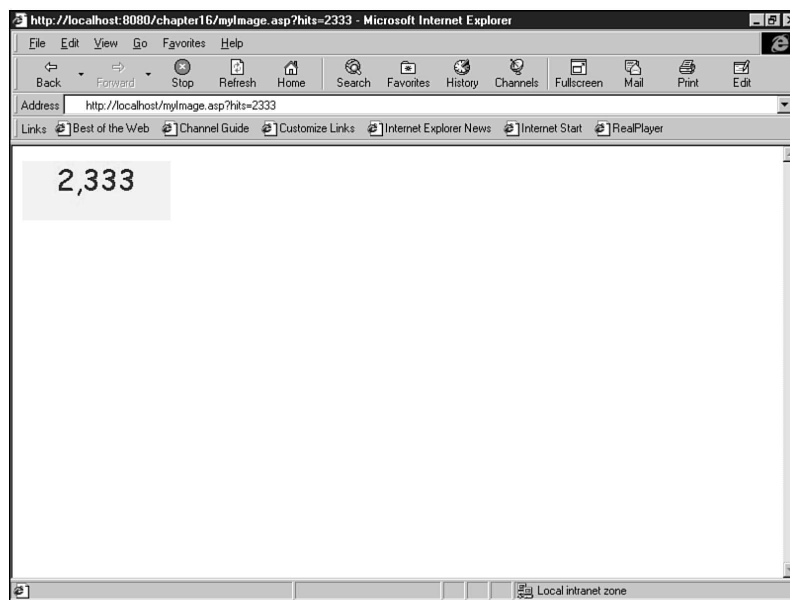

**תרשים 16.1** מונה כניסות טקסטואלי משעמם.

מעניין הרבה יותר יהיה להציג את מספר הבקשות לדף על ידי תמונה, מאשר על ידי טקסט. אם מספר הבקשות מוצג על ידי תמונה, ביכולתך לקבל שליטה מלאה על הצבע והגופן בהם תשתמש כדי להציג את המידע. באפשרותך להשתמש ברכיב ShotGraph כדי ליצור באופן דינמי תמונה בפורמט JPEG, באמצעות התסריט הבא:

```
<%
Response.ContentType="image/jpeg"
hits = Request.QueryString( "hits" )
width = 150
height = 60
Set g=CreateObject("shotgraph.image")
g.CreateImage width, height, 2
'Yellow color
g.SetColor 0, 255, 255, 0
'Blue color
g.SetColor 1, 0, 0, 255
g.SetBgColor 0
g.FillRect 0, 0, width, height
g.SetBkMode "TRANSPARENT"
g.CreateFont "Comic Sans MS", 1, 40, 0, False, False, False, False
g.SetTextAlign "TA_CENTER","TA_BASELINE"
g.SetTextColor 1
g.TextOut width / 2, height /2, FormatNumber( Hits, 0 )
Response.BinaryWrite g.JpegImage( 90,0,"" )
%>
```



שמור את התסריט הבא בשם myImage.asp (נמצא בתקליטור). כאשר תטען את התסריט בדפדפן שלך, יוצגו תמונות שונות בהתאם לערך שתעביר על ידי משתנה מחרוזת השאילתה (Query String Variable) בשם hits. לדוגמה, אם תרשום את כתובת URL הבאה http://localhost/myImage.asp?hits=2333 בסרגל הכתובת של הדפדפן, יופיע הערך 2232 כתמונה.

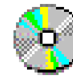



**תרשים 16.2** יצירת תמונה מותאמת עם הרכיב ShotGraph.

התמונה נוצרת על ידי קריאה לשיטה CreateImage השייכת למופע של הרכיב ShotGraph. השיטה ShotGraph מקבלת שלושה פרמטרים: רוחב התמונה, גובה התמונה, ומספר הצבעים בתבנית הצבעים של התמונה. בדוגמה הקודמת, נוצרת תמונה עם תבנית צבעים בעלת שני צבעים.

השיטה SetColor משמשת כדי לקבוע את שני הצבעים בתבנית הצבעים. הפרמטר הראשון של השיטה מציין את אינדקס הכניסה בתבנית הצבעים, ושלושת הפרמטרים הנוספים מכילים את ערכי הצבעים אדום, ירוק וכחול (ערך RGB) של הצבע.

השיטה CreateFont משמשת לקביעת הגופן של הטקסט המוצג. בדוגמה הקודמת, הטקסט מוצג בגופן Comic Sans MS. באפשרותך להשתמש בכל גופן המותקן אצלך במחשב. למעשה, הטקסט מתווסף לתמונה על ידי קריאה לשיטה textOut.



לבסוף, התמונה מוצגת באמצעות הקריאה לשיטה ()BinaryWrite של האובייקט Response. כדי להציג את התמונה שנוצרה באופן דינמי בדף ASP, תוכל להשתמש בתגית HTML הרגילה <IMG>, עם מספר הכניסות שהועבר עם משתנה המחרוזת של השאילתה. התסריט הבא (נמצא בתקליטור בשם FancyHitCounter.asp) מציג בצורה נכונה את התמונה שנוצרה על ידי myImage.asp :

```
<%
Application.Lock
Application( "home_hits" ) = Application( "home_hits" ) + 1
Application.UnLock
 %>
<html>
<head><title>Home Page</title></head>
<body>
Welcome, this page has been requested
<p>
<img src="myimage.asp?hits=<%=Application( "home_hits" )%>">
<p>
times!
</body>
</html>
```

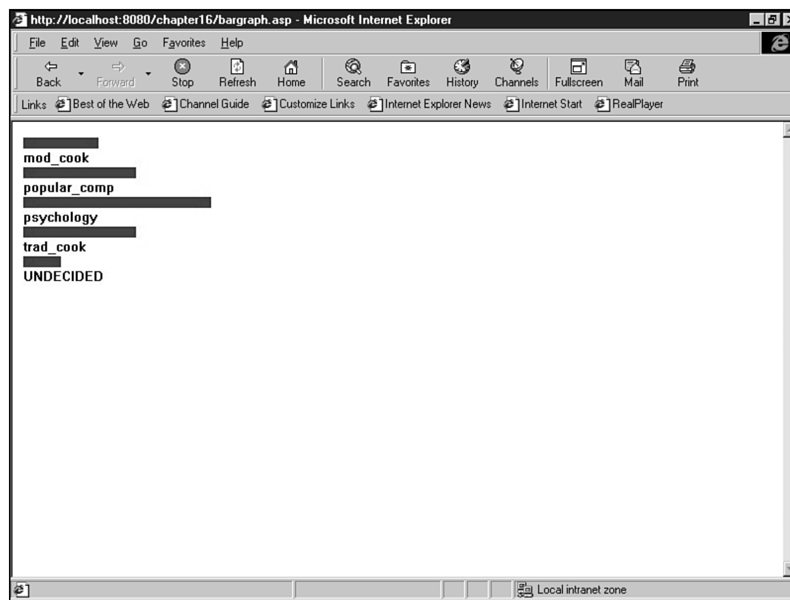
# יצירת גרפים עם הרכיב ShotGraph

אחד השימושים המועילים ביותר של הרכיב ShotGraph הוא ליצירת גרפים. הרכיב יכול לשמש ליצירה דינמית של גרפים מנתונים המאוחסנים בטבלת מסד נתונים. בחלק זה תלמד כיצד לבצע זאת.

כאשר התקנת את שרת SQL, כברירת מחדל הותקן גם מסד נתונים בשם Pubs. מסד נתונים זה מכיל מספר דוגמאות של טבלאות הקשורות לחברת פרסום דמיונית. בתוכנית הדוגמה שלנו נשתמש בטבלה בשם Titles המוכלת בטבלה Pubs. טבלה זו מכילה כותרות של ספרים, ואת הסיווג שלהם (ספרות עסקית, ספרות מחשבים וכדומה).

התסריט בתדפיס 16.1 מציג תרשים עמודות המראה את מספר הספרים מהטבלה Titles מתחת לכל קטגוריה. התסריט מבצע את הפעולות הבאות :

1. רשימת הקטגוריות ומספר הכותרות תחת כל קטגוריה מאוחזר על ידי הצהרת SQL הבאה : select type, count(*) theCount from titles group by type.




**תרשים 16.3** יצירת תרשים עמודות עם הרכיב ShotGraph.

**.2** התוצאות מהשאילתה למסד הנתונים מועברות למערך באמצעות השיטה GetRow(), השייכת לאובייקט RecordSet. דבר זה נעשה כדי לקבוע את מספר המימדים המקסימלי של הגרף, לפני ציורו.

**.3** מספר העמודות שיש לצייר נקבע על ידי אחזור גבול המערך העליון, באמצעות השיטה UBOUND() של VBScript.

**.4** מספר הכותרות המקסימלי מתחת לכל קטגוריה מאוחזר על ידי מעבר על המערך. ערך זה משמש לחישוב אורכה המירבי של כל עמודה בתרשים העמודות, כדי שנוכל לשנות בהתאמה את גודל העמודות, כך שיתאימו לגודל התמונה.

**.5** נוצר מופע של הרכיב ShotGraph ונוצרת תמונה ראשונית ריקה, על ידי קריאה לשיטה CreateImage() עם מימדי התמונה.

**.6** העמודות מצוירות על התמונה, על ידי שליחת הערכים של איברי המערך לשיטה Rectangle().

**.7** התמונה מועברת כפלט לדפדפן, על ידי קריאה לשיטה BinaryWrite() של האובייקט Response.

**תדפיס 16.1** ציור גרף עמודות מטבלת נתונים (BarGraph.asp)

```
<%
Response.ContentType="image/jpeg"
MySQL = "select type, count(*) theCount from titles group by type"
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "DSN=myDSN;uid=sa;pwd=Secret"
SET RS = Con.Execute( MySQL )
myArray = RS.GetRows
' Get Count of Rows
count = UBOUND( myArray, 2 )

' Get Maximum Value
for i = 1 to count
    if myArray( 1, i ) > maxValue then
        maxValue = myArray( 1, i )
    end if
next

' Create ShotGraph Instance
Set g=CreateObject("shotgraph.image")
g.CreateImage 200, 30 * count, 2
g.SetColor 0, 255, 255, 255
g.SetColor 1, 0, 0, 255
g.SetBgColor 0
g.FillRect 0, 0, 200, 30 * count

' Draw The Bars
g.SetBGColor 1
g.CreatePen "PS_SOLID", 1, 1
ypos = 1
for i = 1 to count
    xpos = cINT( myArray( 1, i ) * ( 190 / maxValue ) )
    g.Rectangle 1, ypos, xpos, ypos + 10
    g.TextOut 1, ypos + 12, myArray( 0, i )
    ypos = ypos + 30
next
Response.BinaryWrite g.JpegImage( 90,0,"" )
%>
```

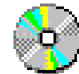
# רכיבים להעברת קבצים

בפרק 3 למדת כיצד לעבוד עם לחצן HTML להעברת (Upload) קבצים. לדוגמה, טופס HTML הבא מכיל לחצן המאפשר למשתמשים לבחור תמונה מכונן הדיסק הקשיח המקומי שלהם, ולהעביר אותה לאתר האינטרנט שלך:

```
<HTML>
<HEAD><TITLE> File Upload </TITLE></HEAD>
<BODY>
```

```
<FORM ENCTYPE="multipart/form-data"
ACTION="upload.asp" METHOD=POST>

Please choose a picture to upload
<BR><INPUT NAME="picture" TYPE=FILE ACCEPT="image/*">
<BR><INPUT TYPE=SUBMIT VALUE="Submit Me !">
</FORM>

</BODY>
</HTML>
```

תוכל להשתמש בשיטות שתוארו בפרק 3 כדי לאחזר ולתרגם את המידע שהועבר. בכל אופן, דבר זה יכול להיות מסורבל מאוד, במיוחד אם אתה חייב לטפל בהעברה של מספר קבצים באותו טופס, או אם ברצונך להשתמש במספר אלמנטים בתוך הטופס. פתרון טוב הרבה יותר הוא להשתמש ברכיב של צד שלישי (Third-Part Components, שפותחו על ידי מי שאינם Microsoft) להעברת קבצים.



מספר רכיבים של צד שלישי שבאפשרותך לקנות יאפשרו לך לטפל בהעברת קבצים. תוכל למצוא באתר האינטרנט המלווה ספר זה, בכתובת www.aspsite.com רשימה של כמה מן הרכיבים הפופולריים ביותר. אחד מהרכיבים הקלים ביותר לשימוש הוא הרכיב SA-FileUp. גרסת התנסות של רכיב זה נמצאת בתקליטור המצורף לספר זה.

### הערה:

גרסת ההתנסות של הרכיב Software Artisans SA-FileUp המצורפת לספר זה מגבילה אותך לטפל בהעברה של לא יותר מאשר 50 קבצים. כדי לרכוש את הגירסה המלאה של SA-FileUp, בקר באתר האינטרנט של Software Artisans, בכתובת http://www.softartisans.com.

ניתן להשתמש ברכיב SA-FileUp כדי לשמור קובץ שהועבר אל הכונן הקשיח בשרת שלך, לשמור את הקובץ למסד נתונים, או רק להעביר את תכולת הקובץ למשתנה. לדוגמה, תוכל להיעזר להשתמש בקוד ASP הבא, כדי לשמור לכונן תמונת GIF שהועברה לאתר באמצעות הטופס שתואר לעיל.

```
<%
Set upl = Server.CreateObject("SoftArtisans.FileUp")
%>
<HTML>
<HEAD><TITLE> File Upload </TITLE></HEAD>
<BODY>
<% IF upl.isEmpty THEN %>
```

```
You didn't enter a file to upload!
<%
ELSE
    upl.Form( "picture" ).SaveAs "c:\temp\theImage.gif"
%>
The image was saved as the file C:\temp\theImage.Gif
<%
END IF
END IF
%>

</BODY>
</HTML>
```

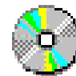
דוגמה זו שומרת קובץ כלשהו שהועבר על ידי המשתמש, בשם c:\temp\theImage.gif. לפני שתוכל להשתמש בדוגמה זו, חובה עליך לוודא שיש למשתמש את ההרשאות המתאימות כדי לשמור קובץ לספריה זו. אם אתה anonymous, תצטרך להעניק את הרשאת המשתמש IUSER_MACHINENAME כדי לכתוב לספריה זו. כדי לשנות הרשאות גישה של משתמש לספריה, לחץ לחיצה ימנית על שם הספריה, מתפריט הקיצור בחר Properties, Security, Permissions.

כאשר משתמשים ברכיב SA-FileUp, חשוב להבין שאין באפשרותך להשתמש עוד באובייקט Request באותו דף. עליך להשתמש ברכיב Upload כתחליף לאובייקט Request, אפילו כאשר אתה עובד עם שדות בטופס אשר אינם שייכים ללחצן העברת הקבצים. לדוגמה, אם היינו משנים את הטופס שתואר בסעיף הקודם כדי שיכיל שדה עבור שם התמונה, היה עליך להשתמש בהצהרה הבאה כדי לאחזר את ערכו של השדה:

```
imageName = upl.Form( "imageName" )
```

# דוגמה ליישום של העברת קובץ

נניח שאתה רוצה לבנות אתר אינטרנט המוקדש לחיות מחמד. ברצונך לכלול דף המאפשר למשתמשים להעביר אליהם תמונות של הכלבים, החתולים או התוכים החביבים עליהם. לאחר שמשתמש העביר תמונה, אתה רוצה שהתמונה תוצג לצפייה בצורה אוטומטית, לכל המעוניין.



כדי ליצור דף זה, באפשרותך להשתמש ברכיב SA-FileUp בצרוף עם הרכיב להעברת קובץ. קוד ASP המופיע בתדפיס 16.2 מאפשר למשתמשים להעביר תמונות (בפורמט GIF או JPEG) לתיקיה באתר האינטרנט הנקראת Photos ולהציג בצורה אוטומטית את כל התמונות בתיקיה:

**הערה:**

כדי ללמוד עוד על הרכיב להעברת קבצים, ראה פרק 8.

**תדפיס 16.2** העברת והצגת תמונות (FILEUPLOAD.ASP)

```
<%
CONST photoDir = "/photos"
Set upl = Server.CreateObject("SoftArtisans.FileUp")
IF upl.userFileName <> "" AND upl.ContentDisposition = "form-data" THEN
    filename = Server.MapPath( photoDir )  & "\" & getFileName( upl.userFileName )
    upl.Form( "photo" ).SaveAs filename
END IF

FUNCTION getFileName( byVal thePath )
    getFileName = RIGHT( thePath, LEN( thePath ) - instrREV( thePath, "\" ) )
END FUNCTION

SUB showPhotos
    Set MyFileObject=Server.CreateObject("Scripting.FileSystemObject")
    Set MyFiles=MyFileObject.GetFolder( Server.MapPath( photoDir ) )
    For each thing in MyFiles.Files
    %>
    <IMG SRC="<%=photoDir & "/" & thing.Name %>"><p>
    <%
    Next
END SUB

%>
<HTML>
<HEAD><TITLE> Photos </TITLE></HEAD>
<BODY>

<% showPhotos %>

<FORM ENCTYPE="multipart/form-data"
ACTION="<%=Request.ServerVariables( "SCRIPT_NAME" )%>"
METHOD=POST>
<HR>
Upload a photo:
<INPUT NAME="photo" TYPE=FILE ACCEPT="image/*">
<INPUT TYPE=SUBMIT VALUE="Upload Photo">
</FORM>

</BODY>
</HTML>
```

כדי להשתמש בדוגמה זו, עליך לשנות את ערכו של הקבוע photoDir לשם הספריה אליה יכול לגשת שרת האינטרנט שלך. כמו כן, תצטרך לשנות את ההרשאות לספריה זו, כדי שלמשתמש בגישה מסוג anonymous על ידי IUSR_MACHINENAME תהיה הרשאת כתיבה לספריה.

# רכיבי רשת

התקשורת בין השרת שלך לשאר שרתי האינטרנט בעולם נעשה באמצעות Sockets. המפרט של Sockets פותח לראשונה עבור Barkeley Software Interface (BSD) על ידי אוניברסיטת קליפורניה שבברקלי, עבור מערכות UNIX. דרך ה-Socket מגיעה מידע ליישום ונשלח מידע לשרת. WinSock (קיצור של **Win**dows **Sock**et) הוא ה-Socket הסטנדרטי במחשבים הפועלים בסביבת Windows.

Winsock מספק ממשק לתכנות יישומים (API), ליצירת ממשק עם פרוטוקול TCP/IP (אף על פי שהוא יכול לעבוד גם עם פרוטוקולים אחרים, כמו למשל IPX/SPX). כיון שכמעט כל התקשורת ברשת האינטרנט מתבצעת באמצעות הפרוטוקול TCP/IP, ה-API של WinSock מאפשר לך גישה ל"צנרת" היסודית של האינטרנט. כלי רב עוצמה!

כדי להשיג גישה ל-API של WinSock דרך ASP, עליך להשתמש ברכיב ASP של צד שלישי (Third-Part Components, שפותחו על ידי מי שאינם Microsoft). קיימים כמה רכיבי WinSock טובים. הבחירה הטובה היא הרכיב w3 socket (ניתן להוריד בחופשיות מהרשת בכתובת http://www.dimac.net) או ASPSock מבית ServerObjects (זמין בכתובת http://serverobjects.com). בפרק זה נשתמש בדוגמה שנציג ברכיב w3 socket.

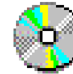
מה בדיוק אתה יכול לעשות עם רכיב WinSock? בעמודים הבאים יוצגו לך שתי דוגמאות. בתחילה, תלמד כיצד להשתמש ברכיב WinSock עם פרוטוקול HTTP, כדי לאחזר ולשלוח דפי אינטרנט. בהמשך, תלמד כיצד להשתמש ברכיב WinSock עם פרוטוקול NNTP, כדי להשיג גישה למאמרים בקבוצות דיון.

## שימוש ב-WinSock כדי לתקשר עם אתרי אינטרנט אחרים

לפני שתוכל להשתמש ברכיבי WinSock כדי לאחזר דפי אינטרנט מאתרי אינטרנט אחרים, הינך חייב להבין את היסודות של פרוטוקול HTTP. כאשר דפדפן שולח בקשה לדף אינטרנט, הבקשה נראית בערך כך:

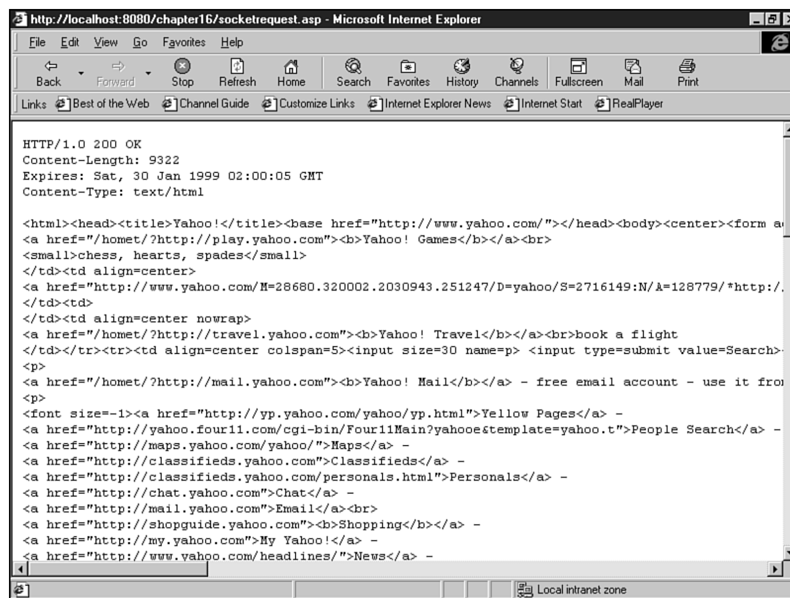
```
GET /example.htm HTTP/1.1
HOST: www.aspsite.com
ACCEPT: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
```

הבקשה הקודמת מאחזרת את הדף example.asp מאתר האינטרנט www.aspsite.com. כאשר השרת באתר www.aspsite.com מקבל את הבקשה, הוא מנסה לאתר את הדף המבוקש, ואז הוא שולח אותו לסוכן (Agent) שביקש אותו (בדרך כלל, הדפדפן שלך).



כדי לבקש דף עם הרכיב WinSock, כל מה שאתה צריך לעשות הוא לשלוח בקשה בפרוטוקול HTTP כדוגמת הבקשה הקודמת, דרך הרכיב. לדוגמה, תסריט ASP הבא (נמצא בתקליטור בשם SocketRequest.asp) מאחזר את דף הבית של Yahoo מהכתובת www.yahoo.com, ומציג את קוד HTML שלו (ראה תרשים 16.4):

```
<%
SET mySock = Server.CreateObject("Socket.TCP")
mySock.Host = "www.yahoo.com:80"
theRequest = "GET / HTTP/1.0" & vbCRLF
theRequest = theRequest & "Accept: image/gif, image/x-xbitmap, image/jpeg,
↳image/pjpeg, */*" & vbCRLF
theRequest = theRequest & vbCRLF
mySock.Open()
mySock.SendLine theRequest
mySock.WaitForDisconnect()
Response.Write "<pre>" & Server.HTMLEncode( mySock.Buffer ) & "</pre>"
mySock.Close()
%>
```




**תרשים 16.4** קוד מקור של דף שהתקבל מרכיב WinSock.

דוגמה זו משתמשת ברכיב w3 Sockets כדי ליצור ממשק עם WinSock API. השורה הראשונה בתסריט יוצרת מופע של הרכיב. בשלב הבא, נקבע המאפיין HOST ל- www.yahoo.com:80, כדי לתקשר עם השרת של Yahoo דרך port מספר 80 (ה-port הסטנדרטי לפרוטוקול HTTP). לאחר שה-Socket חובר ל-port המתאים, נבנית מחרוזת הבקשה. לבסוף, מחרוזת הבקשה נכתבת ל-Socket על ידי הפונקציה SendLine(), ותוצאת הבקשה מוצגת בדפדפן על ידי הצגת תוכן המאפיין buffer של ה-Socket.

## הערה:

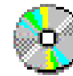
ראה פרק 24, דוגמת תכנות הממחישה כיצד ניתן להשתמש ברכיב WinSock כדי לבדוק בצורה אוטומטית את מצב האתר שלך.

משלוח מידע לטופס באתר אינטרנט אחר הוא תהליך דומה לאחזור דף אינטרנט. כאשר אתה שולח נתוני טופס, לעומת זאת, אתה צריך להשתמש בשיטה POST של פרוטוקול HTTP, ולא בשיטה GET. בנוסף, יש לכלול כותרת CONTENT-TYPE, המציינת את פורמט הנתונים הנשלחים, וכותרת CONTENT-LENGTH המציינת את אורך הנתונים הנשלחים. השיטה POST של HTTP נראית כך:

```
POST /result.asp HTTP/1.1
Content-type: application/x-www-form-urlencoded
Accept: text/plain
Content-length: 33

username:Andrew+Jones&password=secret
```

שים לב שהנתונים שנשלחים מופרדים מכותרות HTTP על ידי שני מופעים של תו Carriage Return (תו המקביל להקשה על Enter) ותו הזנת שורה (Linefeed). בנוסף, המידע מקודד ב-URL. בדוגמה זו, נשלחים שני שדות מהטופס. שדה הטופס הראשון אשר נקרא username, מכיל את הערך Andrew Jones, והשדה השני נקרא password ומכיל את הערך secret. הדפדפן שלך יכול להשתמש בבקשה כדוגמת הבקשה הבאה כאשר ישלח את טופס HTML הבא:

```
<HTML>
<HEAD><TITLE> Simple Form </TITLE></HEAD>
<BODY>
<FORM METHOD="post" ACTION="result.asp">
<BR><INPUT NAME="username" TYPE="text">
<BR><INPUT NAME="password" TYPE="password ">
<BR><INPUT TYPE="submit" VALUE="Submit">
</FORM>
</BODY>
</HTML>
```



באפשרותך לבצע סימולציה של משלוח הטופס הקודם על ידי הרכיב w3 socket. כדי לשלוח את טופס המידע, URL מקודד את המידע לדף הנקרא result.asp בכתובת www.aspsite.com (נמצא בתקליטור בשם postRequest.asp) :

```
<%
postData = "username=" & Server.HTMLEncode( "Andrew Jones" )
postDate = postDate & "&password=" & Server.HTMLEncode( "secret" )
SET mySock = Server.CreateObject("Socket.TCP")
mySock.Host = "www.domain.com:80"
theRequest = "POST  /bin/search HTTP/1.0" & vbCRLF
theRequest = theRequest & "Accept: text/plain" & vbCRLF
theRequest = theRequest & "Content-length: " & len( postData )
theRequest = theRequest & vbCRLF & vbCRLF
theRequest = theRequest & postData
mySock.Open()
mySock.SendLine theRequest
mySock.WaitForDisconnect()
Response.Write mySock.Buffer
mySock.Close()
%>
```

# שימוש ב-WinSock כדי לתקשר עם קבוצות דיון ברשת

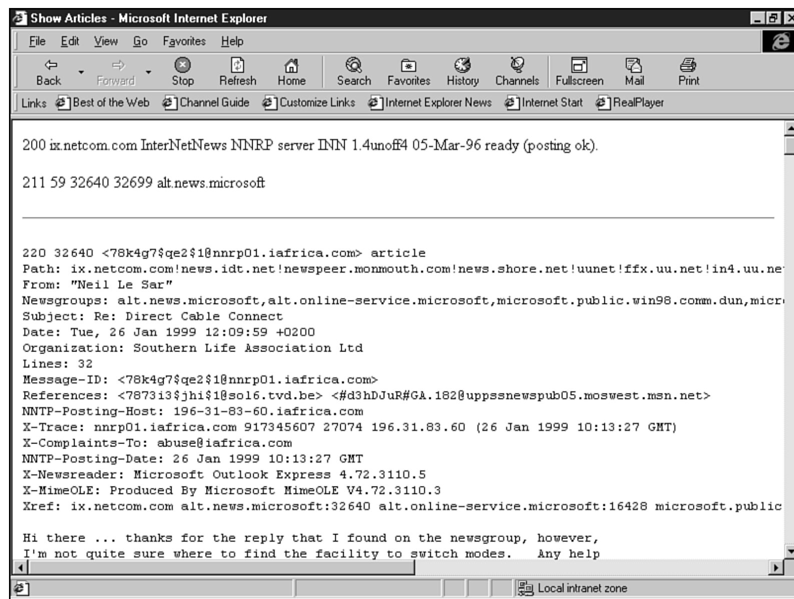
רכיב WinSock יכול לשמש גם כדי לתקשר עם שרת NNTP. פרוטוקול NNTP - פרוטוקול להעברת ידיעות ברשת (Network News Transport Protocol), הוא הפרוטוקול הסטנדרטי למשלוח ואחזור מאמרים מקבוצות דיון באינטרנט כדוגמת microsoft.public.inetserver.iis ו- alt.collection.beanie-babies. ביכולתך גם להקים שרת NNTP משלך בעזרת Microsoft NNTP Server הנכלל ב- Windows NT Option Pack.

הפרוטוקול NNTP, כמו הפרוטוקול HHTP, מורכב מערכת פקודות הניתנות לקריאה על ידי אדם, באנגלית. לדוגמה, כדי לבחור קבוצת דיון מסוימת כמו alt.collection.beanie-babies, יש לשלוח את הפקודה GROUP alt.collection.beanie-babies לשרת NNTP. כאשר השרת מקבל פקודה זו, הוא משיב מידע על מספר המאמרים בקבוצת הדיון ועל מספר המאמר הראשון והאחרון בקבוצת הדיון.

התסריט בתדפיס 16.3 מתחבר לשרת NNTP ומבקש ומציג את כל המאמרים הנוכחיים בקבוצת הדיון alt.news.microsoft (ראה תרשים 16.5). בתחילה, התסריט יוצר חיבור (connection) לשרת NNTP הממוקם ב- nntp.domain.com ב-port 119 (ה-port הסטנדרטי לפרוטוקול NNTP). לאחר החיבור (connection), בוחר התסריט את קבוצת הדיון alt.news.microsoft על ידי הפקודה GROUP של פרוטוקול NNTP. בשלב הבא, מוצגים כל המאמרים בקבוצת הדיון על ידי הפקודות ARTICLE ו-NEXT של פרוטוקול NNTP. הפקודה ARTICLE מאחזרת את המאמר מקבוצת הדיון הנוכחית שנבחרה, והפקודה NEXT עוברת למאמר הזמין הבא בקבוצת הדיון.

### הערה:

Port הוא מספר אשר מקבל כל יישום שרץ על מערכת ההפעלה. מספר זה מזהה למי מיועד המידע הנשלח לאותו מחשב כדי שיגיע ליעדו (היישום). ניתן לחשוב על Port כעל שלוחת טלפון במרכזיה. לדוגמה: אם אנו רוצים לשלוח מידע למחשב מסוים בעל כתובת IP אין זה מספיק רק לציין את כתובת ה-IP שלו, חייבים לציין גם את היישום אליו מיועד המידע באותו מחשב וזהו מספר ה-Port.




**תרשים 16.5** רשימה של קבוצות דיון שאוחזרה באמצעות הרכיב WinSock.

### הערה:

כדי ללמוד בהרחבה אודות הפקודות של פרוטוקול NNTP, בצע חיפוש ב-Yahoo (www.yahoo.com) אחר הערך NNTP protocol. הפרוטוקול מוגדר במסמך RFC 977.

**תדפיס 16.3** בקשת מאמרים מקבוצות דיון (REQUESTNEWS.ASP)

```
<HTML>
<HEAD><TITLE>Show Articles</TITLE></HEAD>
<BODY>
<%
' Connect To News Server
SET mySock = Server.CreateObject( "Socket.TCP" )
mySock.Host = "nntp.yourdomain.com:119"
mySock.Open
response.write mySock.getLine & "<P>"
```

```
' Get Articles From alt.test
mySock.SendLine "GROUP alt.news.microsoft"
result = mySock.getLine
response.write result & "<P>"

' Get Number Of Articles And First And Last
breakit = SPLIT( result )
numArticles = breakit( 1 )
firstArticle = breakit( 2 )
lastArticle = breakit( 3 )

' Display List Of Articles
while LEFT( result, 3 ) <> "421"
    mySock.SendLine "ARTICLE"
    mySock.WaitFor( vbCRLF & "." & vbCRLF )
    response.write "<HR><PRE>" & mySock.buffer & "</PRE>"
    mySock.SendLine "NEXT"
    result = mySock.getLine
wend

' Disconnect
mySock.SendLine "QUIT"

mySock.Close()
%>
</BODY>
</HTML>
```

# רכיבים נוספים

ישנם מאות רכיבים של צד שלישי (Third-Part Components), שפותחו על ידי מי שאינם Microsoft) המיוצרים הן על ידי חברות והן ועל ידי בודדים, בהם תוכל להשתמש להרחבת דפי ASP שלך. יתר על כן, ברובם של רכיבים אלה ניתן להשתמש בחינם. להלן כמה מן הרכיבים היותר מעניינים. ראשית, רשימה של שלושה מן המקורות הטובים ביותר למציאת רכיבים באינטרנט:

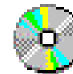
- ASPSite (www.aspsite.com)
- האתר ASPSite הוא אתר האינטרנט השותף לספר זה. בעוד אני כותב שורות אלו, חלק מן האתר המוקדש לרכיבי ASP מכיל רשימה של למעלה מ- 150 רכיבים. הרכיבים מחולקים לקטגוריות וניתנים לחיפוש.
- ActiveServerPages.com (www.activeserverpages.com)
- האתר מכיל רשימה מקיפה של רכיבי ASP. בנוסף, הוא אחד מהמקורות הטובים ביותר באינטרנט על ASP.

- (www.15seconds.com) 15seconds
- 15seconds מכיל חלק שלם המוקדש לרכיבי ASP. בכל אופן, הוא מקור טוב הרבה יותר למאמרים בנושא תכנות ב-ASP.
- (www.serverobjects.com) ServerObjects

ServerObjects היא החברה העומדת מאחורי הרבה מאוד מרכיבי ASP השימושיים ביותר. ל-ServerObjects רכיבי רשת ורכיבים ליצירת תמונות, נוסף על רכיבים ייחודיים רבים נוספים (חלקם יתוארו בהמשך).

# רכיבי גישה לקבצים

בפרק 8 הכרת את הרכיב File Access. הרכיב File Access מאפשר לך לבצע משימות בסיסיות לטיפול בקבצים כמו קריאה וכתיבה של קבצי טקסט, העתקה והעברה של תיקיות. למרבית המשימות הקשורות בקבצים, פועל הרכיב File Access טוב למדי.



יכול להיות שתצטרך לבצע משימות מתקדמות יותר אשר לא יוכלו להתבצע על ידי רכיב זה. התקליטור המצורף לספר זה מכיל גירסה תפקודית מלאה של הרכיב SA-FileManager מבית Software Artisans. רכיב זה תואם לחלוטין לרכיב File Access, ובנוסף מכיל שיטות ותכונות נוספות משלו. לדוגמה, שלא כמו רכיב File Access, תוכל להשתמש ברכיב SA-FileManager כדי להצפין ולפענח קבצים בכונן הקשיח שלך. תכונה זו יכולה לשמש כדי להגן על סיסמאות או על מידע רגיש אחר הנמצא בשרת. בנוסף, מכיל הרכיב גם שיטות לעבודה עם קבצים בינאריים. כדי לראות את התיעוד המלא על רכיב זה, בקר באתר הבית של Software Artisans בכתובת http://www.softartisans.com.

# רכיבי דואר אלקטרוני (E-Mail)

בפרק 10 למדת כיצד להשתמש ב-CDO עבור CDO for NTS) NTS) כדי לשלוח ולקבל דואר אלקטרוני, תוך שימוש בפרוטוקול SMTP. בכל אופן, בצורתו הנוכחית, ל-CDO מגבלות חמורות מסוימות. ראשית, CDO אינו תומך בפרוטוקול POP. משמעות הדבר היא, שאינך יכול להשתמש ב-CDO כדי לאחזר דואר אלקטרוני מתיבות דואר רבות הנמצאות בשרת POP. שנית, וחשוב יותר, CDO אינו עובד בצורה תקינה תחת עומס כבד, הנוצר על ידי משתמשים רבים העובדים בו זמנית, (אם יותר ממשתמש אחד שולח הודעת דואר אלקטרוני באותו זמן, ישנה אפשרות שאחת מההודעות תימחק ולעולם לא תישלח).

כדי לעקוף מגבלות אלו, ביכולתך לרכוש רכיב דואר אלקטרוני של צד שלישי. לדוגמה, הרכיב AspPOP3, מבית ServerObjects, מאפשר לך לתקשר עם שרת POP3. שימוש ברכיב זה מאפשר לך לשלוח את שמו של שרת POP3 שלך, את שם המשתמש (Username), ואת הסיסמה שלך (Password), ולאחזר את כל הודעות הדואר האלקטרוני שלך. כדי ללמוד עוד על רכיב זה בקר באתר http://www.serverobjects.com.

אם ברצונך לשלוח דואר אלקטרוני רב מהאתר שלך, בחן את הרכיבים AspQMail (של ServerObjects בכתובת www.serverobjects.com), ו-ocxQMail (של Flicks Software's בכתובת www.flicks.com). שני רכיבים אלה מאפשרים לך לאגור (Spool) הודעות דואר אלקטרוני כך שהודעות נוספות יוכלו להישלח מייד. ל-ocxQMail יש תכונה מעניינת נוספת, המאפשרת לך לשלוח דואר אלקטרוני באופן אוטומטי בתאריכים מתוכננים. באפשרותך להשתמש בתכונה זאת, לדוגמה, כדי לשלוח הודעת דואר אלקטרוני אוטומטית למשתמש מייד לאחר שנרשם לאתר שלך.

## רכיבים להצגת כרזות פרסומת (Banners)

בפרק 9 למדת כיצד להשתמש ברכיב AD Rotator כדי להציג כרזות (Banners) פרסומת. הרכיב AD Rotator מוגבל להצגת תמונות בלבד. כרזות פרסומת המכילות תסריטי JavaScript הופכות להיות פופולריות יותר ויותר, כך שמגבלה זו הופכת להיות רצינית.

במקום להשתמש ברכיב AD Rotator, באפשרותך להשתמש ברכיב של צד שלישי להצגת כרזות פרסומת, כמו הרכיב AdShark של oceantek (אותו ניתן למצוא בכתובת http://oceantek.com/adshark/features.html). AdShark יכול לשמש להצגת פרסומות המכילות תמונות, HTML, JavaScript, ואפילו Java. בנוסף, לרכיב זה יש גם תכונות תיעוד רחבות, כדי לעקוב אחר מספר הפעמים שכותרת פרסום הוצגה או שמשתמש לחץ עליה. לבסוף, באפשרותך להשתמש ב-AdShark כדי להגביל את תצוגת הפרסומות על ידי מספר ההתרשמויות מהפרסומת, מספר ההעברות, או תאריך.

## רכיבים לבדיקת טפסים (Form Validation)

אם אתה מחייב משתמשים להירשם באתר שלך על ידי מסירת מידע כגון כתובת וכרטיס אשראי, עליך לבדוק את הרכיב SA-Check של Software Artisans. הרכיב SA-Check מאפשר למנוע ממשתמשים לשלוח מידע אקראי כאשר הם שולחים טופס HTML. הרכיב בודק את חוקיות הכתובת, ובכלל זה ערים, ארצות, מדינות, מיקוד, מספרי טלפון וכתובות דואר אלקטרוני. ניתן גם להשתמש ברכיב זה כדי לבדוק חוקיות של מספר כרטיס אשראי שנרשם בטופס, אך הדבר נעשה ללא ביצוע בדיקה מעשית לאימות הכרטיס (המחשב משתמש באלגוריתם 10 "MOD" כדי לקבוע אם מספר הכרטיס שנרשם אכן חוקי מבחינת המבנה). לבסוף, באפשרותך להשתמש ברכיב זה כדי לנפות גסויות אשר משתמשים יכולים לרשום בטופס שבאתר שלך.

## רכיבים לעבודה עם Word ו-Excel

לעיתים דף HTML אינו השיטה המתאימה ביותר להצגת מידע למשתמש. במקום להשתמש ב-ASP כדי ליצור דף אינטרנט סטנדרטי, באפשרותך להשתמש ברכיב של צד שלישי בתסריט ASP שלך, כדי ליצור מסמכי Word או גיליונות אלקטרוניים של Excel.

לדוגמה, Chili!Reports מבית Chili!Soft's (בכתובת www.chilisoft.com) מאפשר לך ליצור באופן דינמי גיליון אלקטרוני של Excel מתוך דף ASP, ולשלוח אותו ברשת. בעזרת Chili!Reports, ביכולתך להשיג גישה לכל הפונקציות, הנוסחאות, טבלאות הצירים והגרפים של Excel, כאשר אתה יוצר את גיליון העבודה.

רכיב Sterling Strategic Solutions Productivity ASP, לא רק יוצר גיליונות אלקטרוניים של Excel, אלא יכול גם לשמש ליצירה דינמית של מסמכי Word. כדי לראות הדגמה של רכיב זה, בקר באתר http://sterlingweb.com.

# סיכום

פרק זה סיפק לך סקירה כללית לגבי מה תוכל לבצע באמצעות רכיבי ASP של צד שלישי. בסעיף הראשון, למדת כיצד להשתמש ברכיב ShotGraph כדי ליצור באופן דינמי תמונות JPEG מתוך דף ASP שלך. בסעיף השני, למדת כיצד להשתמש ברכיב SA-FileUp מבית Software Artisan, כדי לטפל בצורה פשוטה בהעברת קבצים לאתר שלך באמצעות טופס HTML רגיל. בסעיף השלישי, למדת כיצד להשתמש ברכיב WinSock כדי לתקשר עם שרתי HTTP ו-NNTP. לבסוף, הוצגה בפניך סקירה קצרה של כמה רכיבים מהמעניינים יותר, הזמינים להורדה או לרכישה באינטרנט.

## פרק 17

# יצירת Windows Script Components

**בפרק זה:**

- מהו Windows Script Components?
- התקנת Script Component Engine
- היכרות עם XML
- אנטומיה של רכיב תסריט
- עבודה קלה עם אשף רכיב התסריט
- הגדרת המאפיינים של רכיב התסריט
- הוספת שיטות לרכיב התסריט
- שיטות ניפוי וטיפול בשגיאות
- שימוש ברכיב תסריט בדפי ASP

במהלך ספר זה, למדת על השימוש ברכיבים בתוך קבצי ASP. רכיבים אלה מגוונים מאוד, החל ברכיבי ASP אינטגרליים, כגון Request ו-Response, ועד למספר רכיבים של צד שלישי (Third-Party Components) שנכתבו על ידי מי שאינם Microsoft) שתוארו בפרק 16. עתה הגיע הזמן ללמוד כיצד לבנות רכיבים משלך.

אף על פי שמירב הסיכויים הם שניתן למצוא רכיב של צד שלישי שיענה כמעט על כל צורך, לשימוש ברכיבים יכולים להיות גם חסרונות. הראשון והחשוב מכל, אין לך כל שליטה על ההתפתחות העתידית של הרכיב. אף על פי שאולי הרכיב עונה על צרכיך כיום, הוא יכול להשתנות בכל רגע על ידי יצרן הרכיב לרכיב פחות שימושי עבורך. בנוסף, הינך תלוי ביצרן לשם קבלת תמיכה, תיקון של באגים, וכדומה.

החסרון השני של שימוש ברכיב של צד שלישי הוא שקרוב לוודאי שלא תהיה לך גישה לקוד המקור (Source Code) של הרכיב. המשמעות היא, שהרכיב יתפקד לחלוטין כ"קופסה שחורה" שעבודתה תישאר לא נגישה אליך. משמעות נוספת היא, כי לא תוכל לשנות אפילו במעט את התנהגות הרכיב כדי להתאים אותו בצורה טובה יותר לצרכיך.

לאחר ששקלת חסרונות אלה, קרוב לוודאי שהחלטת שיש לך את הזמן, האנרגיה והרצון לבנות רכיבים משלך. פרק זה ושני הפרקים שלאחריו, מתארים כיצד להשתמש בשלושת השיטות הנפוצות ביותר כדי ליצור רכיבים הנגישים מתוך קוד ASP: Windows Script Components, הנושא של פרק זה, Visual Basic, ו- Visual C++. יצירת רכיבים ב- Visual Basic וב- Visual C++ יתוארו בפרקים 18 ו- 19.

בפרק זה תלמד תחילה מהו Windows SCript Component (רכיב תסריט Windows), ומתי תרצה להשתמש ברכיב כזה. מכיון שרכיבי תסריט אינם מותקנים עם Internet Information Server או עם ASP, תצטרך לטעון ולהתקין אותם בעצמך. נושא זה מתואר בסעיף, **התקנת מנוע רכיב התסריט** (Script Component Engine), מאוחר יותר בפרק זה. לאחר שהתקנת את המנוע, תוצג לפניך שפת הסימון המורחבת - XML (eXtensible Markup Language) המשמשת לתיאור רכיב לעולם החיצוני.

לאחר שתרכוש את הידע הבסיסי, תוכל לקרוא בהמשך פרק כיצד להוסיף מאפיינים, שיטות וקוד תסריט לרכיב. אפילו תוכל "להציץ" קצת לתוך הנושא של דיבוג ותיקון שגיאות שאולי יתגלו.

# מהו Windows Script Components?

רכיבי תסריט (Script Components) נקראו בתחילה על ידי Microsoft תסריטונים (scriptlets). המונח תסריטון מרמז שאנו עוסקים בתסריט קטן מאוד, בדומה לאנלוגיה בין Java app לבין Java applet (Java applet, הנקרא גם יישומון, הינו יישום הרץ במחשבי הלקוח. הוא מופעל מתוך דף HTML באמצעות התגית <applet> שאחד ממאפייניה מציין את שם קובץ היישומון עם הסיומת class. . היישומון עבר הידור כשלהוא בטרם הפעלתו על ידי הדפדפן). על פי הגדרה זו, המונח אינו מתאים. עובדתית, Windows Script Component יכול להיות בכל גודל שהוא. לכן, מהו אם כך

רכיב תסריט? באופן פשוט, Windows Script Component הוא רכיב COM הניתן לשימוש חוזר, שביכולתך ליצור על ידי שפת תסריטי ActiveX כלשהי.

מנקודת המבט של העולם החיצוני, רכיב תסריט נראה בדיוק כמו כל רכיב COM/ActiveX אחר. בקבצי ASP, לא תוכל לומר מהו ההבדל בין רכיב Script לבין רכיב Ad Rotator הניתן להתקנה. זהו טבעו של ממשק COM: כל רכיב COM נראה לקוד המשתמש בשירותיו כ"קופסה שחורה". הקוד המשתמש בשירותי הרכיב נקרא **לקוח** (Client), והרכיב עצמו, בשעה שנמצא בשימוש על ידי הלקוח, נקרא **שרת** (Server). הקוד ישתמש ב-Windows Script Component בדיוק כפי שישתמש ברכיב Ad Rotator.

טבעם זה של רכיבי COM כקופסה שחורה הוא המקום שבו הדמיון בין Windows Script Components לבין רכיבים אחרים, מסורתיים יותר, רכיבי COM (שנוצרים בדרך כלל על ידי Visual Basic ו- ++Visual C) מסתיים. הנקודות הבאות מציינות את ההבדלים בין Script Components לבין רכיבים מהודרים (שעברו תהליך הידור):

- רכיבי תסריט הם בדרך כלל קטנים ויעילים. למרות שהם לא יעילים כמו רכיבים מהודרים. רכיב תסריט אופייני כולל שורות קוד מעטות מאוד המוכמסות (encapsulated) לתוך ממשק COM.
- רכיבי תסריט אינם מהודרים בשום דרך או צורה. הם נוצרים בעזרת שימוש במעבד תמלילים טקסטואלי פשוט, בתגיות XML בסיסיות, ובקוד תסריט. לכן, הינך יכול לבחון ולשנות את הרכיב בפרק זמן קצר מאוד.
- רכיבי תסריט דורשים מעט מאוד קבצי תמיכה בשעת ריצה (Runtime support files). אלה כוללים את המנוע לשעת ריצה של רכיב התסריט (SCROBJ.DLL, שיתואר בהמשך), ואת המנוע לשעת ריצה של שפת תסריטי ActiveX שנמצאת בשימוש בתוך התסריט. אם רכיב התסריט משתמש ב-VBScript או ב-JScript בשרת ה-IIS, קבצים אלה כבר מותקנים. רוב הסיכויים הם, שגם אם אתה משתמש בשפות נוספות כמו Python או PERLScript, גם כן התקנת כבר קבצים אלה.

**הערה:**

רכיבים שעברו הידור (קומפילציה) רשומים בקובץ בקוד בינארי. רכיבים אלה הם בדרך כלל רכיבי ActiveX למיניהם.

לראשונה הציגה Microsoft את המושג תסריטון (הנקרא כיום, Windows Script Component) יחד עם Internet Explorer 4.0. לגירסה זו של דפדפן האינטרנט הייתה יכולת להציג לא רק רכיבי ActiveX רגילים, אלא גם סוג חדש של רכיב הנקרא תסריטון DHTML (Scriptlet DHTML). תסריטונים אלה ניצלו את פרוטוקול Dynamic HTML (DHTML) כדי לספק כתיבת תסריטים למודל אובייקט הדפדפן ולעשיית שימוש חוזר באובייקט תצוגה (Encapsulation).

לאחר שהשימוש בתסריטוני DHTML תפס אצל מפתחים אינטרנט רבים, הבינה Microsoft כי לטכנולוגיה זו יכולים להיות יישומים חשובים רבים גם בצד השרת. שימוש ב-Windows Script Components של צד-שרת (server-side) מאפשר למפתח ב-ASP

לארוז קוד תסריט הבא לידי שימוש לעיתים קרובות, לחבילת COM בעלת מראה מוכר. אריזה זו מאפשרת לך לתחזק ולעשות שימוש חוזר בצורה פשוטה בקוד תסריט ה-ASP.

לאחר הקדמה קצרה על האופן בו מאפשר COM להשתמש ב-Windows Script Component בצד השרת, נדון בשאר הפרק באפשרויות ליצור ולהשתמש ברכיבים אלה.

# התקנת Script Component Engine

התקנת תמיכה ל-Windows Script Component ואשף רכיב התסריט היא בוודאי אחת מהמשימות הקלות ביותר שתעשה עם IIS ו-ASP. לאחר שתעביר אליך את קבצי ההתקנה מאתר הבית של microsoft, פשוט הרץ את קובץ ההתקנה. כעת הינך מוכן ליצור ולהשתמש ברכיבי תסריט (למרות זאת, בוודאי תרצה לקרוא קודם את יתר הפרק). סעיף זה מתאר את תהליך ההתקנה.

## העברת המנוע והתקנתו

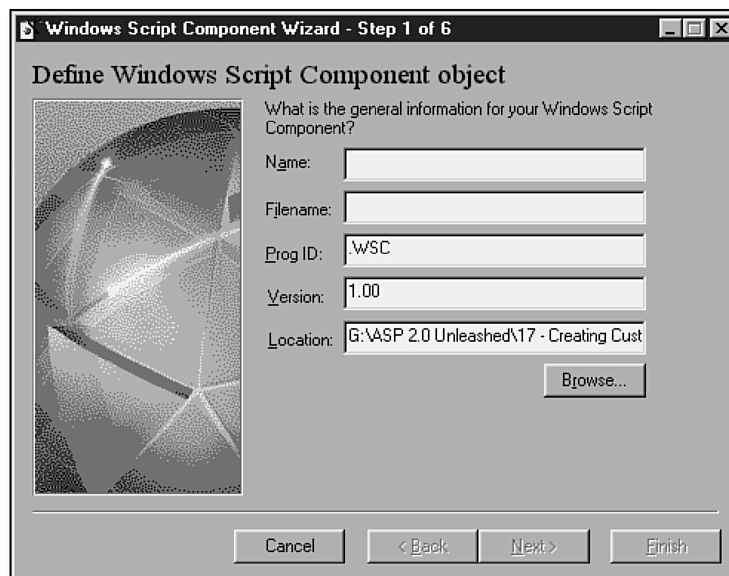
ניתן להוריד את קובץ ההתקנה של מנוע רכיב התסריט, יחד עם הסבר מועיל מאוד בפורמט של קובץ עזרה ב-HTML, מאתר הבית של Microsoft. עבור לכתובת http://www.microsoft.com/scripting ולחץ על הקישור Windows Script Component. כאשר תראה את הקישור לדף ממנו ניתן להוריד את הקבצים, הקש עליו והעבר את הקבצים המתאימים (הורדת אשף רכיב התסריט נפרדת מהורדת יתר המנוע).

לאחר העברת הקובץ למחשב, לחץ לחיצה כפולה עליו, כדי להתחיל את תהליך ההתקנה. עבור דרך תיבות הדו-שיח השונות, ולבסוף תופיע לפניך תיבת דו-שיח המודיעה על סיום ההתקנה. תאמין או לא, אבל זה כל מה שנחוץ. למרבה הצער, יהיה עליך לעבוד עוד קצת כדי לגרום לרכיב התסריט לעבוד!

## מה תמצא כאשר תסיים

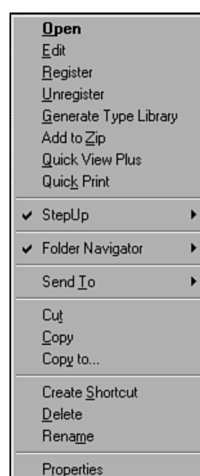
התקנת Windows Script Component ואשף רכיב התסריט מתבססת על המרכיבים הבאים:

- מנוע זמן ריצה (Runtime Engine), SCROBJ.DLL - זהו הממשק בין קובץ ASP לבין קובץ רכיב התסריט. מודל זה אחראי על התקשורת בין ממשק COM של רכיב התסריט לבין הלקוח. דבר זה נעשה על ידי מפרש (Interpreter) הממשק הציבורי של הרכיב (כפי שמוגדר בקובץ הרכיב) ותרגומו לממשק האוטומציה אותו מצפה המשתמש לראות.
- אשף רכיב התסריט (Script Component Wizard) - אשף פשוט אך רב עוצמה זה יסייע לך ליצור את מעטפת (Shell) רכיב התסריט. אני ממליץ לך מאוד ליצור את כל רכיבי התסריט בעזרת אשף זה. תלמד עוד על הדברים שהאשף מאפשר לעשות וכיצד להשתמש בו בסעיף, **עבודה קלה עם אשף רכיב התסריט**. קיצור דרך לאשף זה ממוקם בתפריט התחל (Start), תוכניות (Programs), Microsoft Windows Script. תרשים 17.1 מראה את תיבת הדו-שיח הראשונית של האשף.




**תרשים 17.1** תיבת הדו-שיח הראשונית של Windows Script Component Wizard

- כאשר תלחץ לחיצה ימנית על קובץ WSC בסייר Windows (wsc היא הסיומת האופייניות לקבצי Windows Script Components) ייפתח לפניך תפריט קיצור גדוש בקיצורים שימושיים. תרשים 17.2 מראה את פריטי התפריט הזמינים. פריטי התפריט Open ו-Edit מבצעים את אותה מטלה: הם פותחים את הקובץ בפנקס הרשימות (Notepad). האפשרות Generate Type Library - יצירת ספריית סוג, תיצור ספריית סוג (TLB.) עבור רכיב התסריט. דבר זה עוזר לסביבות פיתוח כמו Visual InterDev ו- Visual Basic להבחין מה זמין ברכיב התסריט. הפריט Register ירשום את הרכיב למערכת ויהפוך אותו לזמין כרכיב COM, והפריט Unregister יסיר את נתוני הרישום מהמערכת.




**תרשים 17.2** תפריט קיצור של **סייר Windows** עבור קובץ רכיב תסריט Windows.

# הקדמה ל-XML

Windows Script Components אינם קבצי ASP. הם אינם קבצי HTML, וגם אינם קבצי include של JavaScript או של VBScript. במקום זאת, הם קבצים בעלי קוד תסריט מובנה, התומכים בממשק ציבורי, ומתועדים בעזרת XML.

XML הוא קיצור של e**X**tensible **M**arkup **L**anguage - שפת סימון מורחבת. קובץ XML דומה לקובץ HTML, אולם יש ביניהם כמה הבדלים חשובים, בהם נדון בסעיף זה. XML יכול לשמש ליצירת מסמך מובנה המבוסס על ערכת חוקים המוגדרים בקובץ הגדרת סוג מסמך - DTD (Document Type Definition). קבצי רכיבי תסריט משתמשים ב-XML כדי לתאר את חלקי הקובץ השונים, כדי לאפשר למודול SCROBJ.DLL למצוא את המידע הדרוש לו כדי לרשום (Register) את רכיב התסריט ולעבוד עם הממשק שלו.

## השוואה בין XML ל-HTML

למרות שקבצי XML ו-HTML נראים דומים, עליך להיות מודע לכמה הבדלים חשובים הקיימים ביניהם. הבדלים אלה קיימים בין אם אתה יוצר קובץ רכיב תסריט, ובין אם אתה יוצר מסמך מובנה המייצג מבנה הירכי של מסד נתונים.

ההבדל הגדול ביותר הוא ש-HTML מאפשר שימוש בערכה קבועה מראש של תגיות (Markup Tags). XML מתוכנן לאפשר לכותבים בו גמישות רבה בהגדרת ערכות פרטיות של תגיות. קובץ DTD מתאר את תחביר התגיות המופיעות במסמך. אתה בעצמך לא תיצור קובץ DTD עבור רכיבי התסריט, מכיון שהקובץ SCROBJ.DLL דורש מבנה מיוחד, אשר קובץ הרכיבים חייב לציית לו.

בנוסף לכך, קיימים שלושה הבדלים חשובים מאוד בין XML ל-HTML שעליך להיות מודע אליהם:

- תגיות XML הינן תלויות רישיות (Case sensitive). דבר זה כולל גם סוגי איברים (כמו <method>) וגם שמות תכונה (כמו Name). דבר זה בעצם אומר שתגית הכתובה כך: <method name="Initialize"></method>, שונה מאוד מתגית הכתובה כך: <METHOD name="Initialize"></method> (תגית שאינה נכונה, דרך אגב).
- ערכי תכונות צריכים להיכתב בין מרכאות או בין גרשיים. לדוגמה, התגית <script language=JScript> היא תגית שגויה. התגית הנכונה היא <script language="JScript">, או <script language='JScript'>.

- תגיות חייבות לבוא תמיד בזוגות, תגית פתיחה ותגית סיום, כמו <script> ו-</script>. ב-HTML ישנו איבר פשוט <BR> שיכול להיכתב בפני עצמו. ב-XML הדבר אסור. תמיד חייב לבוא איבר סיום. לתגיות אשר אין להם תגיות צאצא (Child Tags) ניתן לסגור את התגית על ידי שימוש בלוכסן לפני תו סגירת התגית (</), כמו למשל <method name="Initialize"/>.
- תווים בגוונים מנוגדים המוכלים במרכיבים חייבים להיות בתוך קטע המיועד להיות קטע נתונים (במינוח XML, דבר זה נקרא "opaque"). תחביר זה מונע מתוכנית הניתוח (Parser) לבלבל בין תווים שמורים אלה לבין סימני XML חוקיים. עליך ליצור קטע נתונים (Opaque) סביב לכל קוד התסריט, מכיון שבדרך כלל הוא יכיל תווים שמורים כגון >, <, ו-&. הדבר נעשה על ידי "עטיפה" של קוד התסריט באיבר: <[[ קוד התסריט ]CDATA]!>.

# ציון התאמה ל-XML ברכיב תסריט

מכיון שהמטרה העיקרית של תגיות XML בקובץ Windows Script Component היא לספק ל-SCROBJ.DLL את המידע הדרוש כדי לרשום ולהפעיל את הרכיב, אתה רשאי להתייחס להגבלות שתוארו בסעיף הקודם לא באופן נוקשה מדי. בכל אופן, אם ברצונך לערוך את הרכיב בעורך XML, עליך לדבוק בנוקשות בכללים של שפת XML.

כדי לציין שקובץ הרכיב מציית לכללים של שפת XML, כלול את האיבר <?XML ?> כאיבר הראשון המופיע בקובץ. לאיבר ישנה תכונה הנקראת version שמציינת לאיזו גירסה של XML מתאים הרכיב. התחביר עבור איבר זה הוא:

```
<?XML version="1.0" ?>
```

אם אינך מעוניין שקובץ רכיב התסריט יפורש עם החוקים הנוקשים של XML, פשוט אל תכלול איבר זה בקובץ. אי כתיבת איבר זה בקובץ מאפשרת לך להשתמש בשוני בין התגיות כפי שמרשה שפת HTML.

**אזהרה:**

אם אינך כולל את האיבר <?XML ?> בקובץ, אסור לך לעטוף את קוד התסריט באמצעות האיבר ]CDATA]!> כפי שתואר בסעיף הקודם. אם תעשה זאת תגרום לתוכנית הניתוח להתבלבל, ובכך תגרום לדיווח על טעויות שונות כאשר תרשום, תסיר רישום, או תתפעל את הרכיב.

# האנטומיה של רכיב Windows Script

כעת, לאחר שלמדת מעט על שפת XML וכמה מכלליה, הבה ונראה את חלקה של שפת XML בקובץ רכיב תסריט. כפי שכתבתי בסעיף הקודם, קבצי רכיב תסריט מסודרים כמסמכי XML. משמעות הדבר היא, שהם צריכים להכיל קבוצה מסוימת של איברים, המתארת את הרכיב ואת היכולות הפונקציונליות שהוא מספק. סעיף זה מתאר איברים אלה. מידע נוסף אודות כל אחד מרכיבים אלה ניתן למצוא בתיעוד רכיב התסריט, באתר הבית של MSDN, בכתובת:

http://msdn.microsoft.com/scripting/default.htm?/scripting/scriptlets/doc/lettitle.htm

## האיברים <package> ו- <component>

אם אתה בקי בתכנות COM, אתה בוודאי מודע לכך שרכיב COM בודד יכול להכיל מחלקות COM רבות. הדבר נכון גם לגבי רכיב תסריט. באפשרותך לכלול מספר רב של אובייקטים נפרדים בקובץ רכיב תסריט אחד. הדבר נעשה על ידי שימוש באיברים <package> ו- <component>.

האיבר <package> משמש כדי לסגור על מספר איברים מסוג <component>. כל איבר <component> מייצג אובייקט רכיב תסריט בודד. אם הקובץ מכיל רק אובייקט רכיב תסריט אחד, האיבר <package> הוא אופציונלי.

להלן דוגמה לשימוש באיברים אלה יחד:

```
<package>
<component id="Scriptlet1">
    (component body goes here)
</component>
<component id="Scriptlet2">
    (component body goes here)
</component>
</package> element> element> element> element>
```

## מידע הרישום (Registration Information)

תת מערכת האוטומציה של COM עבור מערכות הפעלה מבוססות Windows מתבסס על מידע המאוחסן ברישום המערכת (Registry) של Windows, כדי לעזור לו לאתר ולבחון את מודל זמן הריצה (Runtime Model) של רכיב COM נתון. רכיבי תסריט אינם יוצאי דופן מכלל זה.

חלק ממבנה XML של רכיב התסריט הוא האיבר <registration>. איבר זה מכיל את כל המידע שצריך להירשם ברישום המערכת על ידי SCROBJ.DLL לשם קבלת גישה לרכיב התסריט מלקוח COM. מידע זה מוכל בתכונות progid, classid, description, ו-version שיתוארו בהמשך.

לפניך מוצגת דוגמה לתחביר:

```
<registration progid="unleashed.wsc"
    classid="{abb298d0-b213-11d2-8580-00a024166737}"
    version="1.00"
    description="Sample Script Component">
</registration>
```

## התכונות progid ו-classid

לקוח COM עושה שימוש בשתי התכונות, progid ו-classid, כדי לציין איזה שרת COM אמור להיות מופעל באותו רגע. לקוח COM ישתמש באחת משתי תכונות אלו כאשר ייצור את השרת (השרת הוא רכיב ה-COM המופעל). שים לב שלפחות תכונה אחת מתכונות אלו חייבת להימצא באיבר <registration> של הרכיב.

התכונה progid, כפי שבוודאי ראית בתחביר הפשוט שהוצג בסעיף הקודם, היא מחרוזת המורכבת בדרך כלל משתי מילים או יותר, המופרדות בנקודה. בדרך כלל, מתארת המחרוזת את מטרתו של הרכיב. מחרוזות PROGID בוודאי כבר מוכרות לך, מפני שהשיטה CreateObject של שרת ASP משתמשת בהם. לצורך הדוגמה הקודמת, היית משתמש בתחביר:

```
set scptlt = Server.CreateObject("unleashed.wsc")
```

התכונה progid היא אופציונלית, אולם אם אינך משתמש בה עליך לציין במקומה את התכונה classid. לקוחות יוכרחו להשתמש בתכונה classid, כדי לזהות את רכיב התסריט.

האיבר classid מתאר את ה-CLSID עבור רכיב התסריט. CLSID הוא מזהה ייחודי גלובלי (GUID) לאובייקט COM. מכיון שמובטח ש-GUID יהיה ייחודי בכל פעם שהוא נוצר, ניתן להשתמש במזהים אלה כדי לזהות רכיב בצורה ייחודית כאשר הוא מותקן במחשבים רבים. מכיון שהערך של progid הוא בעל תבנית הניתנת לקריאה על ידי אדם, קיימת האפשרות שאותו ערך יופיע ברכיבים נוספים, דבר שיהפוך לבלתי אפשרי עבור מערכת ה-COM לדעת בדיוק איזה רכיב יש להפעיל כאשר התכונה progid מצוינת בקריאה לשיטה CreateObject.

ל-Microsoft יש פורמט ופונקציות די מורכבות ליצירת מזהים של CLSID, נושא הנמצא מחוץ לתחומו של דיון זה. מספיק רק לומר שב-Visual Studio קיימים כלים הזמינים בחינם, אשר ייצרו עבורך ערכי CLSID. בדומה, אשף רכיב התסריט ייצור ערך CLSID חדש בכל פעם שתיצור רכיב תסריט. אם אתה משתמש בתגית <OBJECT> בקבצי HTML או ASP, הרי שכבר ראית את CLSID בפעולה.

דוגמה לשימוש ברכיב התסריט שתואר קודם בקובץ ASP עם התגית <OBJECT> היא:

```
<OBJECT id=cmpnt classid="clsid: abb298d0-b213-11d2-8580-00a024166737">
</OBJECT>
```

התכונה classid היא אופציונלית, אולם עליך לציין תכונה זו או את התכונה progid. אם לא תציין את התכונה classid, ייווצר מזהה CLSID בכל פעם שרכיב התסריט יירשם למערכת. הגם שנראה שזהו מכניזם החוסך זמן, הוא יכול בוודאי להוביל לבעיות, מפני שרכיב התסריט יקבל ערך CLSID שונה בכל פעם שיירשם למערכת. משמעות הדבר היא שכל קוד העושה שימוש בתגית <OBJECT> כדי ליצור מופע של רכיב התסריט, יצטרך לעבור שינוי בכל פעם שרכיב נירשם למערכת. מאחר ואשף רכיב התסריט הוא "נדיב" למדי, ויוצר עבורך את ערכי CLSID, הדבר הטוב ביותר הוא להשתמש בהם!

**הערה:**

תחביר ערכי התכונות progid ו-classid אינו תלויות רישיות (Case Sensitive). לכן, גם הערך Unleashed.WSC וגם הערך unleashed.wsc מתייחסים לאותו קובץ רכיב תסריט.

## מידע על הגירסה והתיאור של רכיב התסריט

בנוסף לתכונות progid ו-classid, עליך לציין גם מידע על הגירסה והתיאור של רכיב התסריט.

לקוחות COM הדורשים מופע תלוי-גירסה של השרתים אותם הם יצרו משתמשים בתכונה version. בדרך כלל, מבנה מחרוזת זו הוא n.nn ועליה לגדול בכל פעם שאתה משנה את הממשק הציבורי של רכיב התסריט. COM ישתמש במספר הגירסה כדי לנסות ולאתר מידע אודות הממשק, הקיים עבור מספר גירסה מסוים. לכן, כאשר לקוח ירצה להשתמש בגירסה מסוימת של ממשק, יציין קוד הלקוח את מספר הגירסה המבוקש, ו-COM ינסה לאתר את היישום המתאים לממשק ולגירסה זו. לדוגמה, ניתן לקבוע את הערך x = Server.CreateObject("ADODB.Connection.1.5") כדי לאחזר את גירסה 1.5 של האובייקט Connection של ADO.

ספריית הסוג (Type Library) עושה שימוש בתכונה Description. זוהי מחרוזת אשר עוזרת לזהות את הרכיב לדפדפני אובייקטים, כדוגמת דפדפן האובייקטים הנמצא ב-Visual InterDev וב-Visual Basic. נראה כי הגירסה החופשית של מחולל ספריית הסוג מתעלמת בסך הכל מטקסט זה. במקום זאת, היא מעדיפה תיאור כללי יותר של ספריית סוג התסריטון (Scriptlet Type Library) לכל רכיב תסריט.

## ביצוע תסריט במהלך תהליך הרישום

על אף העובדה, שמנוע זמן הריצה (Runtime Engine) ‎-SCROBJ.DLL אחראי לרשום את המידע הנחוץ המתאר את רכיב התסריט לתוך מערכת הרישום של Windows (Registry), יהיו בוודאי מקרים בהם תרצה שהרכיב יגלה ויגיב לאירוע זה. תוכל לעשות דבר זה בדיוק, על ידי הוספת האיבר <script> בתוך האיבר <registration>.

מנוע זמן הריצה יכול לקרוא לשתי פונקציות בתוך קטע תסריט זה: Register ו-Unregister. להלן דוגמה לאופן בו ניתן להשתמש בתכונה זו:

```
<registration progid="unleashed.wsc"
    classid="{abb298d0-b213-11d2-8580-00a024166737}"
    version="1.00"
    description="Sample Script Component">
<script language="VBScript">
<![CDATA[
function Register()
    MsgBox "Unleased Script Component Registered!"
end function
function Unregister()
    MsgBox "Unleased Script Component has been Unregistered!"
end function
]]>
</script>
</registration>
```

# האיבר <public>

האיבר <public> מכיל תיאור של הממשק הציבורי המסופק על ידי רכיב התסריט. תיאור זה כולל את המאפיינים (Properties), השיטות (Methods), והאירועים (Events) המתגלים על ידי הרכיב. האיבר "עוטף" את האיברים <property>, <method>, ו-<event> המתארים את המאפיינים, השיטות והאירועים האינדיבידואליים הנתמכים על ידי רכיב התסריט. בסעיפים הבאים תלמד לעומק על איברים אלה, אך תחילה מוצגת דוגמה לתחביר:

```
<public>
    <method name="Initialize"/>
    <property name="LastName"/>
    <event name="ItemSaved"/>
</public>
```

# האיבר <implements>

האיבר <implements> מאפשר לרכיב לקבל גישה לפונקציונליות נוספת של ממשקי COM. לדוגמה, האובייקטים המובנים של ASP Request, Server, Session, Application, ו-Response נהפכים זמינים לרכיב התסריט דרך ממשק נפרד.

התחביר המלא עבור האיבר <implements> הוא:

```
<implements type="COMHandlerName" [id="internalName]" [default=fAssumed]>
handler-specific information here
</implements>
```

התכונה type מפרטת את שם הממשק אותו יש ליישם, התכונה id משמשת ליצירת מרחב שמות נפרד בתוך רכיב התסריט (במקרים בהם קיימת התנגשות במתן השמות בין הממשק הנוכחי לממשק אחר שמיושם), והתכונה default מציינת האם ערך התכונה id מתקבל כאשר מראי מקומות לאובייקטים מסופקים על ידי הממשק (ברירת המחדל היא true, וברוב המקרים לא תציין תכונה זו).

כדי להפיק תועלת מהאובייקטים המובנים של ASP, ציין את האיבר <implements> ברכיב התסריט:

```
<implements type="ASP" id="ASP"/>
```

אחרי כן, בתוך קוד התסריט, אתה יכול פשוט לאזכר את האובייקטים המובנים כאילו היית כותב את הקוד בתוך קובץ ASP. דהיינו, פשוט השתמש בקוד הדומה לזה:

```
Session("IsLoggedIn") = 0
```

או:

```
sReturn = Server.HTMLEncode(Session("LastName")).
```

למידע מעמיק יותר על איבר זה, ראה את תיעוד רכיב התסריט (Script Component Documentation) באתר הבית של Microsoft MSDN.

# איברים נוספים הנמצאים ברכיב תסריט

אמנם האיברים הקודמים הם איברים הכרחיים לחלוטין אותם חייב כל תסריט לכלול, אולם מספר איברים נוספים יכולים להועיל לך בכתיבת הרכיבים.

# האיבר <object>

האיבר <object> מתנהג בתוך רכיב תסריט בדיוק כפי שמתנהגת התגית <OBJECT> בקבצי HTML ו-ASP: הוא נועד להגדיר וליצור מופע של אובייקט, שיהיה נגיש לקוד התסריט. השימוש באיבר <object> הופך את האובייקט לנגיש בצורה גלובלית, ומאפשר לכלים לכתיבת תסריטים כדוגמת Visual InterDev לספק השלמת משפטים (Statement completion) לאיברי האובייקט.

תחביר האיבר <object> הוא:

```
<object id="objID" [classid="clsid:GUID" | progid="progID"]/>
```

הערך objID משמש כדי לספק לקוד תסריט שם שיכול לשמש אותו כדי לזהות את האובייקט. שם זה חייב להיות ייחודי במשך כל רכיב התסריט. האיברים classid ו-progid (אתה חייב להשתמש באחד מהם) מציינים את ה-CLSID וה-ProgID לאובייקט COM שנוצר.

# האיבר <resource>

אם תכנתת אי פעם ב- Visual C++ (ואולי ב-Visual Basic) בוודאי השתמשת בקובץ משאבים (Resource File המוכר גם כ-RC File). קובץ זה מאפשר לך לאחסן בתוכו נתונים שרצוי היה שלא יוכנסו לקוד, כמו למשל מחרוזות שצריך לתרגם לשפה שונה מאנגלית.

מנוע זמן הריצה מספק פונקציה בשם getResource, בה משתמש התסריט כדי לאחזר את ערכו של משאב. התחביר לאיבר זה הוא:

```
<resource id="resourceID">
    text or numeric value
</resource>
```

דוגמה לשימוש במשאב מוצגת בתדפיס 17.1.

**תדפיס 17.1** השימוש באיבר <resource>

```
<resource id="errNonDateValue">
    The value passed was not a valid date.
</resource>
<script language="VBScript">
<![CDATA[
Function GetDateInOneWeek(baseDate)
    If IsDate(baseDate) then
        GetDateInOneWeek = DateAdd("w", 1, baseDate)
    Else
        GetDateInOneWeek = getResource("errNonDateValue")
    End If
End Function
]]>
</script> element>> element>
```

# האיבר <reference>

זהו אחד מהאיברים השימושיים ביותר מבין האיברים השונים. על ידי השימוש באיבר <reference> באפשרותך לתת לקוד רכיב התסריט גישה לקבועים המסופקים על ידי רכיבי ספריית סוג אחר של COM. דבר זה שימושי במיוחד אם רכיב התסריט ניגש למודל אובייקט ADO של Microsoft, אשר גדוש בהגדרות של קבועים.

התחביר לאיבר זה הוא :

```
<reference [object="progID" | guid="typelibGUID"] [version="version"]/>
```

באיבר זה, מציינת התכונה object את ה-ProgID של האובייקט אשר הממשק שלו מיוצג על ידי ספריית הסוג אליה יש להתייחס. התכונה GUID מציינת את ה-GUID של ספריית הסוג (לכל ספריית סוג יש מזהה ייחודי, בדיוק כפי שיש לאובייקט COM). עליך לציין רק אחת משתי תכונות אלו. התכונה version משמשת לציין את מספר הגירסה של ספריית הסוג.

לדוגמה, כדי להשתמש בקבועים המשויכים לאובייקט Connection של ADO, עליך לכלול התייחסות לספריית הסוג של האובייקט :

```
<reference object="ADODB.Connection"/>
```

**הערה:**

על האיבר <reference> להופיע בתוך האיבר <component>. אם יש בתוך חבילת התוכנה יותר מרכיב תסריט אחד, ספריית הסוג מתייחסת אך ורק לרכיב התסריט שהאיבר <component> שלו הוצהר.

## האיבר <comment>

האיבר <comment> משמש כדי לסגור טקסט בתוך איברים אחרים של XML, שאתה רוצה שמנוע זמן הריצה של רכיב התסריט יתעלם ממנו. תיעוד רכיב התסריט יכול להוות סיבה טובה להשתמש באיבר זה.

התחביר הוא :

```
<comment>
   text or data here
</comment>
```

# אשף רכיב התסריט

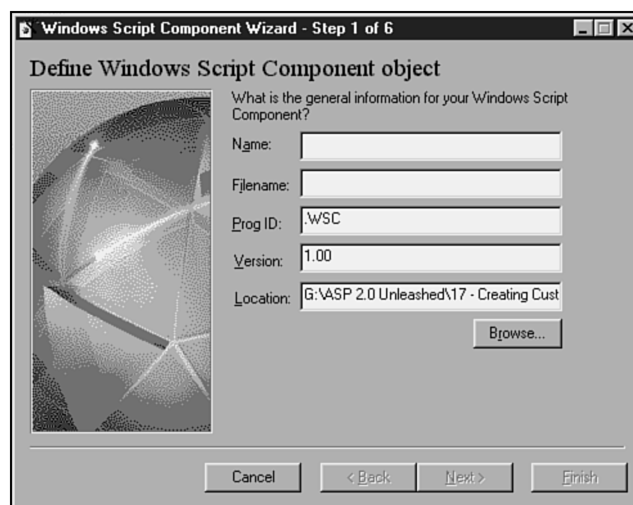
אם הרגשת מבולבל מן הסעיף הקודם, שים לב - על ידי השימוש באשף רכיב התסריט (Windows Script Component Wizard) באפשרותך למכן את רוב יצירת קוד ה-XML בקובץ רכיב התסריט. דבר זה יאפשר לך להתרכז בעבודה האמיתית : כתיבת הקוד של מאפייני ושיטות הרכיב. סעיף זה מתאר כיצד להשתמש באשף.

כדי להפעיל את האשף, לחץ על Start, הצבע על Programs, Microsoft Windows Script, ובחר Windows Script Component Wizard. כדי להשתמש באשף בצע את השלבים הבאים:

**.1** בתיבת הדו-שיח הראשונה של האשף, כפי שמוצגת בתרשים 17.3, תמלא מידע בסיסי על רכיב התסריט. מידע זה יימצא באיבר <registration> של קובץ הרכיב.

השדה Name ימופה לתכונה description. את הטקסט שתמלא בשדה זה ימלא האשף באופן אוטומטי בשדות FileName ו-ProgID. מובן שבאפשרותך להחליף את טקסט ברירת המחדל שבשדות אלה בטקסט משלך.

אם יש צורך, שנה את ערך השדה Version ואת ערך השדה Location המציין היכן יאוחסן קובץ הרכיב. לחץ Next כדי להמשיך.



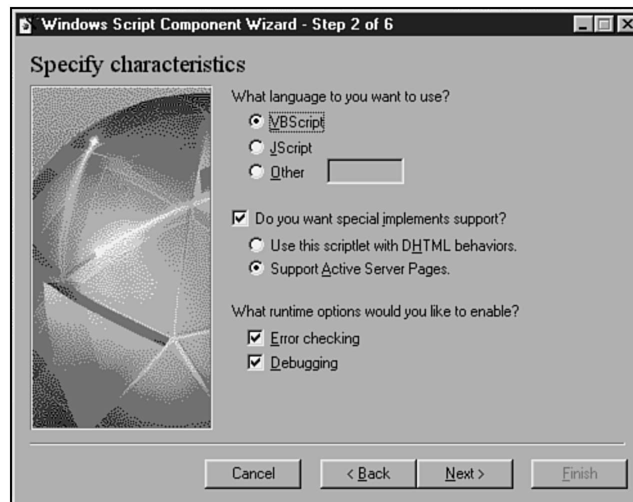

**תרשים 17.3** תיבת הדו-שיח הראשונית של Windows Script Component Wizard

**.2** בתיבת הדו-שיח השנייה, כפי שמוצגת בתרשים 17.4, ציין את מאפייני רכיב התסריט. באמצעות הקבוצה הראשונה של לחצני האפשרויות, ציין את השפה בה תכתוב את התסריט.

באפשרותך לבחור ב-VBScript, JScript או בשפה אחרת. במקרה האחרון תצטרך למלא את מציין השפה. ברכיב התסריט אינך מוגבל להשתמש אך ורק בשפה שבחרת כאן. האשף יוסיף "בדלי קוד" כדי שתוכל לנצל את השפה שצוינה.

כדי לציין שברצונך לקבל גישה לאובייקטים המובנים של ASP, סמן את תיבת הסימון ?Do You Want Special Implements Support, ולחץ על לחצן האפשרויות Support Active Server Pages.

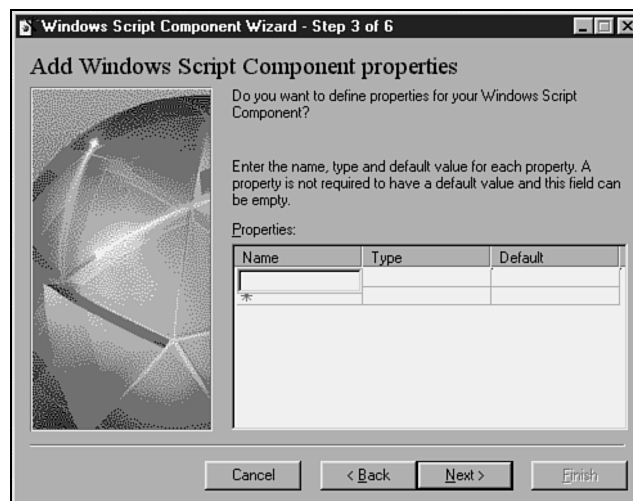
אם ברצונך לאפשר תמיכה בלכידה וניפוי שגיאות ברכיב התסריט, סמן את תיבות הסימון המתאימות. בכל אופן, עליך לאפשר תכונות אלו רק כאשר אתה מפתח את הרכיב, מפני שבשלב ההפקה תרצה שהרכיב ירוץ במצב "שקט". פרטים נוספים על אפשרויות אלו תוכל למצוא בסעיף, **שיטות ניפוי (Debugging) וטיפול בשגיאות (Error-handling)**. לחץ Next להמשך.




**תרשים 17.4** תיבת הדו-שיח Specify Characteristics של Windows Script Component Wizard

**3.** בתיבת הדו-שיח השלישית, המופיעה בתרשים 17.5, עליך לציין את המאפיינים אותם יספק רכיב התסריט.

בעמודה הראשונה בטבלה מלא את שם המאפיין. העמודה השנייה, Type, היא המקום בו עליך לציין האם הרשאת המאפיין היא Read/Write, Read-Only, או Write-Only. לבסוף, העמודה השלישית נועדה לציין ערך ברירת מחדל למאפיין. זהו הערך שיקבל המאפיין כאשר רכיב התסריט ייווצר בפעם הראשונה. לאחר שהשלמת את מילוי המידע אודות מאפייני הרכיב, לחץ Next להמשך.




**תרשים 17.5** תיבת הדו-שיח Add Windows Script Component properties.

4. בתיבת הדו-שיח שתופיע, עליך לציין לאשף אילו שיטות יספק רכיב התסריט. שיטות מיועדות להיות פעולות המבוצעות על ידי קוד הלקוח על הנתונים המאוחסנים באובייקט. בעמודה הראשונה הקלד את שמות השיטות. בעמודה השנייה עליך לציין את הפרמטרים של כל שיטה. הפרד בין שמות הפרמטרים בפסיק, וזכור כי אין ציון של הגדרות סוגים בכתיבת שיטות תסריט (הכל מועבר כנתונים מסוג Variant, מה שאומר שהפרמטר יכול להכיל כל סוג של נתון). אינך צריך לסגור סוגריים מסביב לרשימת הפרמטרים, מפני שהאשף יעשה זאת עבורך. לחץ Next להמשך.

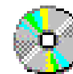
5. תיבת הדו-שיח החמישית היא המקום בו עליך למלא את האירועים אותם אתה מעוניין שיפעיל רכיב התסריט. מכיון שדפי ASP אינם יכולים להגיב לאירועים שנוצרו על ידי האובייקטים הפעילים זמנית, מירב הסיכויים שתצטרך להשאיר מסך זה ריק. אם אתה מתכנן רכיב לשימוש עם Visual Basic או סביבת פיתוח דומה, עליך למלא בתיבת הטקסט את האירועים המופעלים על ידי הרכיב. לחץ Next להמשך.

### אזהרה:

בתיבת הדו-שיח Events, היזהר למלא כל אירוע בשורה נפרדת בתיבת הטקסט. האשף מוסיף שמות אירועים לממשק על ידי שימוש בכל הטקסט שמולא בכל שורה.

6. זה הכל! תיבת הדו-שיח האחרונה מציגה סיכום של קובץ רכיב התסריט שהאשף עומד ליצור. לחץ Finish כדי ליצור את הקובץ.

האשף יודיע לך על יצירת קובץ רכיב התסריט. כעת, באפשרותך להקיש לחיצה כפולה על הקובץ כדי לפתוח אותו בפנקס הרשימות.



תדפיס 17.2 מציג רכיב תסריט פשוט שנוצר על ידי האשף. קוד זה מצורף לתקליטור בשם unleashed.wsc בתיקיה של פרק זה. לרכיב תסריט זה יש שני מאפיינים, LastName ו-FirstName, ושיטה אחת, Initialize. שים לב שרוב העבודה המעיקה הושלמה כבר. כל שנשאר לעשות הוא למלא את הקוד לתמיכה במאפיינים ובשיטות המצויות ברכיב. זהו הנושא של המשך הפרק.

**תדפיס 17.2** רכיב בסיסי שנוצר על ידי אשף רכיב תסריט Windows.

```
<?xml version="1.0"?>
<component>

<registration
    description="Unleashed"
    progid="Unleashed.WSC"
    version="1.00"
    classid="{a5a670a0-f128-11d2-a60b-444553540000}"
>
</registration>
```

```
<public>
    <property name="LastName">
        <get/>
        <put/>
    </property>
    <property name="FirstName">
        <get/>
        <put/>
    </property>
    <method name="Initialize">
        <PARAMETER name="InitFirstName"/>
        <PARAMETER name="InitLastName"/>
    </method>
</public>

<implements type="ASP" id="ASP"/>

<script language="VBScript">
<![CDATA[

dim LastName
dim FirstName

function get_LastName()
    get_LastName = LastName
end function
function put_LastName(newValue)
    LastName = newValue
end function

function get_FirstName()
    get_FirstName = FirstName
end function

function put_FirstName(newValue)
    FirstName = newValue
end function

function Initialize(InitFirstName,InitLastName)
    Initialize = "Temporary Value"
end function

]]>
</script>

</component>
```

# הגדרת המאפיינים של רכיב התסריט

כעת, לאחר שלמדת כיצד להשתמש באשף רכיב תסריט Windows כדי לעשות את רוב עבודת הממשק עבורך, הבה ונצעד צעד אחד לאחור, ונבדוק מדוע עשה האשף מה שעשה.

הגדרות המאפיינים ממוקמות בתוך האיבר <public> (ראה תדפיס 17.2 מוקדם יותר בפרק). על ידי שימוש במאפיינים, קיימות שתי דרכים לחשוף את הנתונים שבתוך הרכיב לעולם החיצון. הדרך הראשונה היא פשוט לחשוף (בפני הרכיב) משתנה מקומי בתוך האיבר <property>. הדרך השנייה היא לעטוף את הנתונים עם הפונקציה get (כדי לקרוא את ערך המאפיין) והפונקציה put (כדי להציב ערך למאפיין).

## שימוש במשתנים כבמאפיינים

אם לא אכפת לך להגן על משתני האובייקט מהעולם החיצון, באפשרותך ליצור מאפיין המקושר ישירות למשתנה פנימי. בכל פעם שקוד לקוח ישנה את ערך המאפיין, ישונה בהתאם גם המידע המאוחסן במשתנה הפנימי. ומעבר לכך, הרכיב שלך אף לא יוכל לזהות שהנתון השתנה.

כדי שעובדות אלו לא ירתיעו אותך, ואם אתה זקוק לדרך פשוטה כדי לחשוף נתונים מסוימים כמאפיינים, להלן התחביר באמצעותו ניתן לחשוף (בפני הרכיב) משתנה מקומי:

```
<public>
    <property name="LastName"/>
    <property name="FirstName" internalname="strFirstName"/>
</public>
```

הגדרת המאפיין הראשון מקשרת מאפיין בשם LastName למשתנה פנימי שגם שמו LastName. ההגדרה השנייה מקשרת מאפיין בשם FirstName למשתנה פנימי בשם strFirstName.

משתנים אלה צריכים להיות מוגדרים כמשתנים גלובליים רק בתוך אחת מתגיות התסריט <script> (בהנחה שהתסריט מכיל יותר מתגית <script> אחת, מה שלא בהכרח חייב להיות):

```
<script language="VBScript">
Dim strFirstName
Dim LastName
וכן הלאה
</script>
```

שים לב לכך שכל מאפיין המוגדר באופן זה הוא מאפיין Read/Write. אינך יכול לשלוט בגישה למאפיין בשום דרך שהיא.

# הגדרת מאפיינים על ידי שימוש בפונקציות

אם האזהרות המאיימות של הסעיף הקודם, בדבר הסכנה בשימוש במשתנים ישירות כמאפיינים, לא השפיעו עליך, הנה כמה מהיתרונות השימוש בפונקציות לאחזור ולקביעת ערכי מאפיין :

- הפונקציה יכולה להוסיף קוד אימות, הבודק את חוקיות הנתונים, לפני קביעה או החזרה של ערך מאפיין.
- באפשרותך להחליט האם המאפיין יהיה Read/Write, Read-Only או Write-Only.

כדי להשתמש בפונקציות לצורך אחזור או קביעת ערכי מאפיין, עליך להוסיף איברים מתאימים מסוג <get> או <put> בתוך איבר המאפיין <property>. האיבר <get> משמש לציון הפונקציה שתיקרא כאשר לקוח מבקש את ערך המאפיין. האיבר <put> מגדיר את הפונקציה שתיקרא כאשר לקוח קובע ערך למאפיין.

כדי לציין מאפיין מסוג Read-Only, פשוט אל תבצע האיבר <put>. כדי לציין מאפיין מסוג Write-Only, אל תכלול את האיבר <get>.

להלן דוגמה מן הסעיף הקודם ששוכתבה לשימוש בפונקציות :

```
<public>
   <property name=”LastName">
      <get/>
      <put/>
   <property name="FirstName"/>
      <get internalname="getFirstName"/>
      <put internalname="putFirstName"/>
   </property>
</public>
```

עבור המאפיין LastName, לא ציינו את התכונה internalname עבור האיברים <get> ו-<put>. במקרה זה עליך ליצור בתוך התסריט פונקציות בשם get_LastName ו-put_LastName. מנוע זמן הריצה מוסיף אוטומטית לשם המאפיין את התחילית _get ו-_put.

עבור המאפיין FirstName, ציינו את שמות הפונקציות לשימוש. פונקציות אלו חייבות להיות מוגדרות עתה גם בקוד התסריט.

קוד הפונקציה get חייב להחזיר את ערכו של המאפיין, כערך החזרה של הפונקציה. הפונקציה חייבת להיות מוגדרת כך שתקבל פרמטר יחיד. ערכו של פרמטר זה הוא הערך אותו הלקוח מנסה לקבוע למאפיין.

# הוספת שיטות לרכיב התסריט

פשוט הרבה יותר להגדיר שיטות מאשר מאפיינים. אינך צריך לדאוג האם המאפיין הוא Read/Write, או האם הוא חושף משתנה בצורה ישירה. כדי להגדיר שיטה, כל שעליך לעשות הוא למקם את האיבר <method> בתוך האיבר <public> של הרכיב. התחביר עבור איבר זה הוא:

```
<method name="externalName" [internalName="internalName"]>
```

הערך externalName הוא שמה של השיטה כפי שמופיע בקוד הלקוח. הערך של התכונה האופציונלית internalName הוא שמה של השיטה בתוך הרכיב. אם שני השמות זהים, באפשרותך להשמיט את התכונה internalName.

אם השיטה דורשת פרמטרים כלשהם, מציינים זאת בתוך האיבר <method> על ידי שימוש באיבר <parameter> אחד לכל פרמטר נידרש, כפי שמוצג בדוגמה הבאה:

```
<method name="externalName" [internalname="internalName"]>
    <parameter name="parameterName">
</method>
```

# Error-handling ו- Debugging

בדרך כלל, רכיבי תסריט של צד השרת "לא יצעקו" בעת ביצוע שגיאה. כאשר הינך מפתח רכיב תסריט, יכול להיות שתרצה להפעיל אותו במצב ניפוי שגיאות (Debug). קרוב לוודאי שתרצה גם לקבל הודעות שגיאה מפורטות יותר.

כדי להשתמש בשני כלים שימושיים אלה קיים איבר אותו באפשרותך להוסיף לקוד הרכיב. תחביר איבר זה הוא:

```
<?component error="errorvalue" debug="debugvalue"?>
```

כברירת מחדל, כל שגיאה בזמן ריצה או שגיאת תחביר תגרום להצגת הודעת שגיאה כללית ביותר. אם ברצונך להפעיל הודעות שגיאה מפורטות יותר, קבע לתכונה error את הערך true.

כמו כן, כברירת מחדל אינך יכול להשתמש בתוכנת ניפוי שגיאות (כמו Visual InterDev) כדי לנפות שגיאות ברכיב תסריט בזמן ריצה. כדי לעשות זאת בכל אופן, עליך לקבוע לתכונה debug את הערך true. תוכנת ניפוי השגיאות תופעל כתגובה לנקודת עצירה (Breakpoint) שנקבעה בסביבת הפיתוח (דבר האפשרי ב-Visual InterDev) או כתגובה להצהרה Stop של VBScript או להצהרה debugger של JScript.

# שימוש ברכיב תסריט בדפי ASP

עתה, כאשר אתה יודע כל שעליך לדעת על יצירת רכיב תסריט, סעיף אחרון זה של הפרק יראה לך כיצד לנצל את הרכיב. מובן, שאם הגעת לפרק זה הינך יודע כיצד להשתמש ברכיב תסריט בדפי ASP, שכן רכיבי תסריט הינם בדיוק כמו אובייקטים אחרים של COM. אך עדיין, קיימים מספר צעדים הנחוצים להעברה של קובץ רכיב תסריט מתוך פנקס הרשימות/שלב התכנון, לשלב היישום.

## הוספת רכיב התסריט לרישום המערכת של Windows

לפני שתוכל להשתמש ברכיב תסריט חדש עם לקוח COM כלשהו, עליך לרשום את הרכיב ברישום המערכת (Registry) של Windows. כפי שהוזכר קודם לכן, תהליך הרישום מוסיף מידע לתוך רישום המערכת של Windows המאפשר למערכת ההפעלה להפעיל רכיב שרת COM.

הרישום ממפה את המידע CLSID וגם/או את ProgID של הרכיב ליישום הממשי של השרת. במקרה של רכיב תסריט, היישום מטופל על ידי מנוע זמן הריצה SCROBJ.DLL, אשר בעזרת שימוש במידע נוסף השמור ברישום המערכת, קורא ומנתח את קובץ רכיב התסריט המתאים.

כדי לרשום רכיב תסריט, לחץ לחיצה ימנית על קובץ רכיב התסריט בסייר Windows ומתפריט הקיצור בחר Register. זה כל מה שנדרש! לאחר שהרכיב נרשם לא תצטרך לרשום אותו שנית, אלא אם כן משהו השתנה בחלק <registration>. כמו כן, אם ביצעת שינוי כלשהו בקוד התסריט המיישם את הרכיב, לא תצטרך להפעיל מחדש את IIS כדי להשתמש בקוד החדש.

אם תעביר את הרכיב למחשב חדש, המלצתי היא להתקין את רכיבי מנוע זמן הריצה של Windows Script Component, כפי שעשית במחשב המקורי.

## יצירת קובץ ספריית סוג

צעד נוסף שתרצה לבצע אם אתה מתכנן להשתמש ברכיב תסריט מקוד שכתבת ב-Visual InterDev, הוא ליצור ספריית סוג (Type Library). כדי ליצור ספריית סוג, לחץ לחיצה ימנית על קובץ קובץ רכיב התסריט ומתפריט הקיצור בחר Generate Type Library. כעת, ייווצר בספריה הנוכחית קובץ TLB. בעל שם זהה לזה של קובץ רכיב התסריט (בזמן כתיבת ספר זה, קובץ ה-TLB. נקרא scriptlet.tlb, אולם אני מצפה שהדבר יתוקן בגירסה הבאה של מנוע זמן הריצה של רכיב התסריט, (Script Component Runtime.

קובץ ספריית הסוג מאפשר ל- Visual InterDev לקבוע אילו שיטות ומאפיינים נתמכים על ידי רכיב התסריט. בעזרת מידע זה יכול InterDev לספק לך השלמת פקודות ו-IntelliSense, שהוא, קרוב לוודאי, החידוש הטוב ביותר בתחום התכנות מזה זמן רב!

# יצירה ושימוש במופע של רכיב תסריט

כמו כל רכיב COM, ASP מספק שתי דרכים ליצירת מופע חדש של רכיב תסריט: בעזרת השיטה CreateObject או התגית <OBJECT>. אם תשתמש בשיטה CreateObject, תציין את הערך ProgID. אם תשתמש בתגית <OBJECT>, תצטרך להשתמש בערך classid של התסריטון (Scriptlet) מתוך האיבר <registration>.

לאחר שנוצר מופע של אובייקט רכיב תסריט, השימוש בו נעשה כבכל אובייקט אחר. באפשרותך לקבוע ולאחזר ערכים של מאפייני האובייקט (בהנחה שהמאפיין מאפשר את הגישה שאתה מנסה לקבל), ולקרוא לשיטות האובייקט.

תדפיס 17.3, נמצא בתקליטור המצורף תחת השם scriptcmp.asp בתיקיה של פרק זה, מדגים דף ASP פשוט המשתמש ברכיב התסריט שנוצר בפרק זה.

**תדפיס 17.3** קובץ ASP פשוט המשתמש ברכיב תסריט

```
<%@ Language=VBScript %>
<OBJECT RUNAT=SERVER ID=scptPet
 CLASSID="clsid:a5a670a0-f128-11d2-a60b-444553540000">
</object>
<%
   set scptOwner = server.CreateObject("unleashed.WSC")
   scptOwner.FirstName = "Craig"
   scptOwner.LastName = "Eddy"

   scptPet.FirstName = "Lucy"
   scptPet.LastName = "Eddy"
%>
<HTML>
<BODY>
<H2>Pet</H2>
First Name: <%=scptPet.FirstName%><BR>
Last Name: <%=scptPet.LastName%><P>
<H2>Owner</H2>
First Name: <%=scptOwner.FirstName%><BR>
Last Name: <%=scptOwner.LastName%><BR>
</BODY>
</HTML>
```

# סיכום

פרק זה הציג בפניך דרך פשוטה ורבת עוצמה כדי ליצור רכיבי COM ממשיים תוך שימוש בכתיבת תסריטים בקוד ActiveX. רכיבים אלה קטנים, קלים לתחזוק, ואינם דורשים מהדר או סביבת פיתוח כדי ליצור אותם.

הפרק הבא יורד לעומקו של נושא יצירת הרכיבים באמצעות Visual Basic. אמנם רכיבים אלה מהודרים, וסביבת הפיתוח Visual Basic צריכה להיות מותקנת במחשב, אולם תוכל להשיג רבות בעזרתם.

## פרק 18

# יצירת רכיבים באמצעות Visual Basic

**בפרק זה:**

- **יתרונות Visual Basic לפיתוח רכיב**
- **יצירת רכיבי צד-שרת (Server-Side)**
- **כתיבת קוד הרכיב**
- **השימוש ברכיב מדפי ASP**
- **היכרות עם היישום WebClass של Visual Basic**
- **יצירת יישום WebClass**
- **עבודה עם WebItems**
- **תגובה לאירועים**
- **טיפול בקלט מטופס**
- **תכונות מיוחדות של יישומי WebClass**

פרק 17 עסק בשימוש בכתיבת תסריטים ב-ActiveX כדי ליצור רכיבי צד-שרת. רכיבים אלה נגישים לקוד דפי ASP באותו אופן כמו האובייקטים המובנים Request ו-Response. ההבדל העצום בכל אופן, הוא שאתה הוא זה שצריך לשלוט על המתרחש ברכיב שיצרת.

אף על פי ששימוש בכתיבת תסריטים ב-ActiveX כדי ליצור רכיבים היא שיטה רבת עוצמה ובעלת יתרונות, שימוש בסביבת פיתוח ייעודית כמו Visual Basic או Visual C++, פותח עולם חדש של הזדמנויות. ישנם כמובן מספר חסרונות ליצירת רכיבים על ידי שימוש בשפה מהודרת, אולם הם מתגמדים לחלוטין לעומת היתרונות שבשיטה זו.

פרק זה עוסק ביצירת רכיבים על ידי Visual Basic, ויעסוק בנושאים הבאים:

- יתרונות השימוש ב-Visual Basic לעומת תסריטונים (Scriptlets) או Visual C++.
- כיצד ליצור רכיבי צד-שרת של ActiveX לשימוש מתוך קוד ASP.
- כיצד להשתמש ברכיבי WebClass של Visual Basic 6.0 כדי ליצור יישום מלא מבוסס IIS, עם Visual Basic.

**הערה:**

פרק זה מספק מספיק פרטים כדי ליצור רכיבים מותאמים עם Visual Basic, אך למרות זאת הוא אינו מתעמק בכל מורכבות כתיבת הקוד של רכיבים אלה באמצעות Visual Basic. נושאים אלה נלמדים באמצעות הספר **Visual Basic 6.0 סדנת לימוד,** בהוצאת **הוד-עמי**.

# יתרונות Visual Basic לפיתוח רכיב

ישנם יתרונות רבים לפיתוח רכיבים מותאמים עם Visual Basic. למרות שתסריטונים מספקים אמצעים קלים ויעילים ליצירת תסריטים, אין להם סביבת פיתוח מלבד אשף התסריטון (Scriptlet Wizard) המסופק על ידי Microsoft, דבר המגביל אותך ליכולות של השפה בה בחרת לכתוב את התסריט.

באופן דומה, מספקת Visual C++ לרכיב שלך כמות עצומה של יכולות פונקציונליות. השימוש ב- Visual C++, הופך את הביטוי "השמיים הם הגבול" כמעט למדויק. באפשרותך לפתח כל סוג של רכיב המבצע כל מטלה נדרשת, במהירות מסחררת. אולם מחיר הגמישות בא על חשבון המורכבות. ללא כל ספק, C++ היא אחת השפות הקשות ביותר לשליטה.

Visual Basic מספקת בסיס אמצעי נהדר לפיתוח רכיבים, במיוחד לרכיבי צד-שרת שאינם דורשים ממשק משתמש (Visual C++ היא במידה רבה השפה הנבחרת ליצירת יישומים מרובי ממשקים). הנה מספר יתרונות לשימוש ב-Visual Basic (כל היתרונות הללו, למעט האחרון, מתייחסים לגירסה 5.0 או 6.0):

- זו שפה פשוטה הניתנת לשליטה בנקל.
- Visual Basic מאפשרת עבודה פשוטה ביותר עם סוגי הנתונים הבסיסיים. עבודה עם מחרוזות ב- C++ יכולה להיות משימה קשה מאוד, עקב הייצוגים הזמינים הרבים שבה. כמו כן, השימוש במצביעים (Pointers) בטוח יותר ב-Visual Basic. ברוב המקרים, אפילו לא תבחין שאתה עובד עם מצביע. ב- C++ לעומת זאת, עליך לנהוג בזהירות רבה מאוד בעבודה עם מצביעים.
- Visual Basic מספקת סביבת עבודה מלאת תכונות, עם תכונות כגון IntelliSense, השלמת משפטים, וכלים אינטגרליים לעבודה עם מסדי נתונים.
- מהדר הקוד המקומי משתמש באותו מהדר כמו Visual C++. משמעות הדבר היא, שבאפשרותך לבנות רכיבים מהירים שגודלם ניתן לשינוי.
- דפדפן האובייקטים (Object Browser) מאפשר לך לראות את ממשקי הרכיבים כפי שצוינו בקוד.
- Visual Basic 6.0 מכניסה לראשונה לשימוש את WebClass. זהו סוג מיוחד של פרויקט שתוכנן להפעלה על Internet Information Server (IIS), גירסה 3.0 ומעלה. WebClass הוא יישום IIS מושלם, הכולל את היכולת להשתמש בקבצי תבניות HTML, במקום בקוד חוזר ונשנה של Response.Write. בחלקו השני של פרק זה נדון ב-WebClass בפרוטרוט.

# יצירת רכיבי צד-שרת (Server-Side)

אם קראת ספר זה עד לנקודה זו, בוודאי הבנת שרוב העבודה הנעשית על ידי דפי ASP מושגת על ידי רכיבי ActiveX (הקרויים גם רכיבי COM) הפועלים על שרת (או על מכונה אחרת, אם אתה משתמש ב-DCOM). האובייקטים Request ו-Response, שנראים מובנים ב-ASP, הם בעצם רכיבי COM המופעלים עבור הקוד שלך על ידי IIS.

השימוש ב-Visual Basic (מגירסה 4.0 והלאה), מאפשר לך ליצור בקלות ערכת רכיבים משלך. בדומה לאובייקטים Request ו-Response, יהיו רכיבים אלה נגישים מתוך דפי ה-ASP שלך. ההבדל היחיד הוא, שלמרות שנראה שהאובייקטים Request ו-Response הם רכיבים מובנים, רכיבים שאתה יוצר צריכים להיות מופעלים על ידי קוד ה-ASP.

Visual Basic עושה את רוב העבודה השחורה המשויכת לבניית רכיבי ActiveX/COM עבורך. עליך רק להגדיר את מודול אובייקט הרכיב (זוהי ערכת מאפיינים גלויים ושיטות גלויות), לכתוב את הקוד, ולהדר. Visual Basic תוסיף עבורך את כל פונקציות ה-COM ברמה הבסיסית. פונקציות אלו משמשות את מערכת ההפעלה כדי לרשום את הרכיב שלך ולהפעיל את המאפיינים והשיטות שלו. בדיוק כמו בשימוש במנוע זמן הריצה (Runtime Engine) של התסריטון שתואר בפרק 17, השימוש ב-Visual Basic משחרר אותך מהעבודה השחורה.

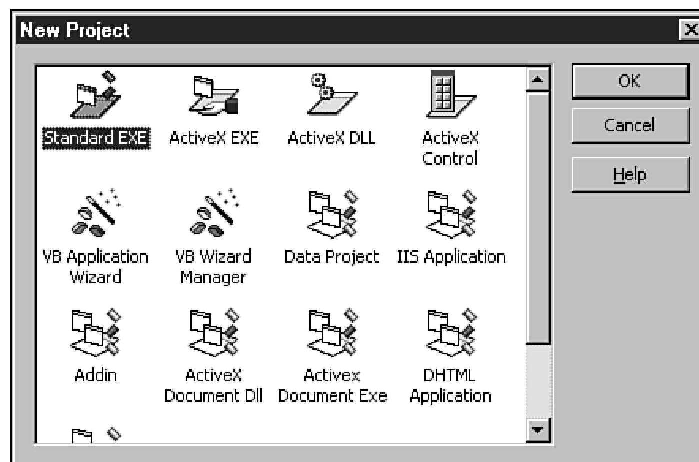
**הערה:**

פרטים נוספים על בניית רכיבי IIS/ASP עם Visual Basic תמצא בכתובת:

http://msdn.microsoft.com/workshop/server/components/vbmtsiis.asp

# יצירת הפרויקט

השלב הראשון ביצירת רכיב מותאם עם Visual Basic הוא ליצור פרויקט ActiveX חדש. ל-Visual Basic היכולת להדר את הקוד לסוגים שונים של קבצי "הפעלה" (Executable). כאשר אתה מפעיל את סביבת הפיתוח של Visual Basic, או כאשר אתה בוחר מהתפריט File באפשרות New Project, תעמוד בפניך האפשרות ליצור כל אחד מסוגים אלה ומסוגי פרויקטים נוספים (ראה תרשים 18.1).



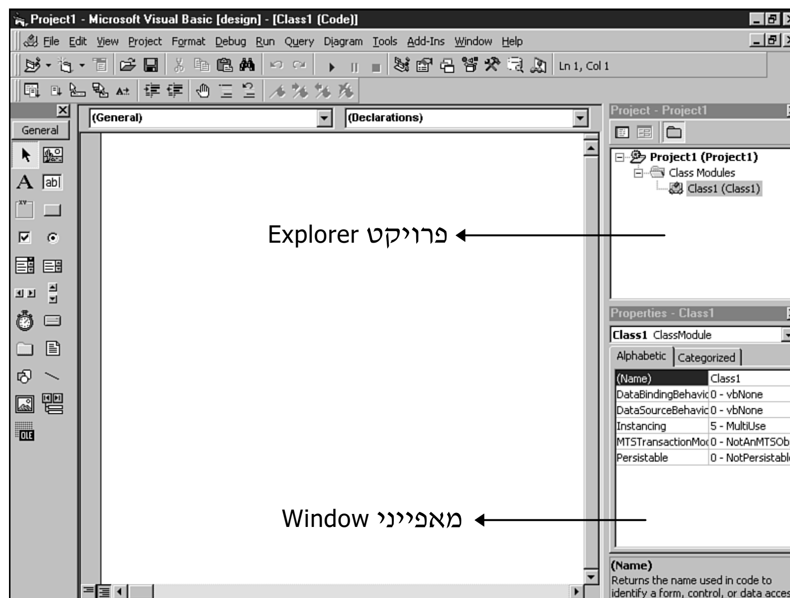

**תרשים 18.1** תיבת הדו-שיח New Project של Visual Basic.

- קבצי Win32 EXE סטנדרטיים - אלה הם יישומי Windows סטנדרטיים.
- קבצי ActiveX EXE - אלה הם רכיבי שרת של ActiveX/COM הפועלים מחוץ לתהליך (process). כלומר, הם פועלים בתהליך אחר מאשר התהליך בו פועל קוד הלקוח שהפעיל את הרכיב. כל יישום מופעל בתהליך שניתן לו על ידי מערכת ההפעלה. למרות שניתן לגשת לקבצי ActiveX EXE מדפי ASP, דרך זו אינה מומלצת ודורשת שינוי בנתוני-על של IIS, המאפשרים הפעלה של כל קבצי ההפעלה מתוך קוד תסריט הפועל בשרת. לפרטים נוספים, ראה את התיעוד של IIS.
- קבצי ActiveX DLL - אלה הם רכיבי ActiveX/COM הפועלים באותו תהליך (process) כמו הלקוח שהפעיל את הרכיב. בדרך כלל, הרכיבים המותאמים שתיצור יפלו תחת קטגוריה זו.
- יישומי IIS - זהו סוג מיוחד של יישום העושה שימוש ב-HTML כדי להציג את ממשק המשתמש שלו, ובקוד מהודר של Visual Basic כדי לעבד בקשות לתגובות לאירועים מהדפדפן. יישומים אלה יתוארו בחלקו השני של פרק זה.
- בקרי ActiveX (OCX) - אלה הם רכיבים מיוחדים הכוללים את התמיכה הנוספת הדרושה ליישום מארח, כדי להציג ולטפל הרכיב בסביבת ממשק משתמש גרפי - GUI. בנוסף, באפשרותך לשבץ רכיבי OCX בדף אינטרנט. בדרך כלל, רכיבי OCX הם רכיבי ממשק-משתמש, ואינם יכולים להיווצר על ידי השימוש בקריאה ל-Server.CreateObject. לכן, לא תיצור הרבה רכיבים כאלה לשימוש בקוד ASP.
- מסמכי ActiveX DLL ומסמכי ActiveX EXE - מסמך ActiveX מאפשר לך להציג טפסי Visual Basic בתוך יישומי מכולת מסמכי OLE (OLE Document Container), כמו ב-Internet Explorer או ב-Microsoft Office Binder. למרות שזהו רעיון נהדר, אין לו שום יישום בדפי ASP ולכן לא יתואר כאן.

**הערה:**

קבצי DLL ו-EXE של ActiveX מאוזכרים לעיתים כ-"רכיבי קוד", כיון שבדרך כלל הם משמשים כרכיב בלתי נראה המספק שירותים שונים דרך קוד, בניגוד לאינטראקציית משתמש.

כדי ליצור רכיב נגיש-ASP, לחץ לחיצה כפולה על הסמל ActiveX DLL בתיבת הדו-שיח New Project של Visual Basic. Visual Basic ייצור פרויקט חדש עם מודול מחלקה בודד (ראה תרשים 18.2).




**תרשים 18.2** תחילתו של רכיב ActiveX DLL.

הפרויקט מייצג את הרכיב. כברירת מחדל משתמשת Visual Basic בשם Project1 לפרויקט. מובן שתרצה לשנות שם זה לשם בעל משמעות כלשהי. כל מודול מחלקה מייצג אובייקט בודד בתוך הרכיב. מודול המחלקה הראשוני נקרא, כמה מפתיע, Class1. שוב, ודאי תרצה לשנות שם זה.

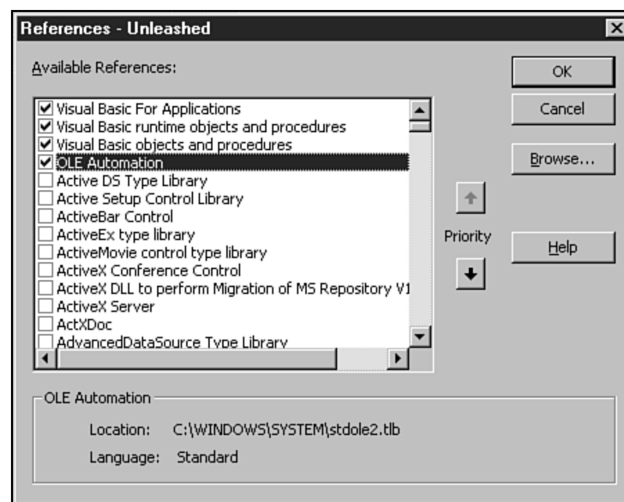
היכן שמות הפרויקט והמחלקה משחקים תפקיד? נסה להיזכר בתחביר השיטה Server.CreateObject. ודאי אתה זוכר כי השיטה מקבלת כפרמטר מחרוזת, שהיא ה-ProgID של הרכיב שיופעל. ProgID משמש את מערכת ההפעלה לאתר את קובץ ההפעלה שיספק את המופע הפעיל של הרכיב. ProgID הוא בדרך כלל במבנה servername.objectname. במקרה של Visual Basic, החלק servername הוא שם הפרויקט והחלק objectname הוא שם המחלקה.

לפיכך, אם תחליט להשאיר את שמות ברירת המחדל של Visual Basic ללא שינוי, ערכו של ProgID יהיה Project1, וקוד ה-ASP שבאמצעותו ייווצר המופע של אובייקט זה יהיה Server.CreateObject("Project1.Class1"). ברור שמצב זה אינו מתקרב אפילו לתיעוד עצמי נורמלי של הקוד, ולכן קרוב לוודאי שתרצה לבחור לפרויקט ולמחלקה שמות אשר יתארו את תפקידו ומטרתו של הרכיב שאתה יוצר. כדי לשנות את השמות, לחץ על הפריט (פרויקט או מחלקה) בעץ שבחלון סייר הפרויקט (Project Explorer, אם הוא אינו נראה, הקש Ctrl+R). לחץ על המאפיין **(Name)** בחלון Properties, ושנה את ערכו. אם החלון Properties אינו נראה, הקש F4.

# גישה לאובייקטים מובנים של ASP

אחד השימושים הטובים ביותר לרכיבים מותאמים שניגשים אליהם מתוך דפי ASP הוא להחליף קוד שנעשה בו שימוש בקבצי ASP רבים. כדי לעשות זאת, תצטרך לגשת לאובייקטים מובנים של Request) ASP, Response, Server, Session ו-Application). למרבה המזל, בדיוק כמו בתסריטוני שרת, מסייעת Visual Basic לבצע מטלה זו.

השלב הראשון בגישה לאובייקטים מובנים הוא להוסיף ייחוס לספריות הסוג המתאימות לפרויקט שלך. יהיה עליך להוסיף ייחוס ל-Microsoft Active Server Pages Object Library ול-Microsoft Transaction Server Type Library. כדי לעשות זאת בסביבת הפיתוח של Visual Basic, פתח את תפריט Project ובחר References. תוצג תיבת הדו-שיח References, כפי שמוצג בתרשים 18.3.




**תרשים 18.3** תיבת הדו-שיח References של Visual Basic.

גלול את הרשימה Available References עד למציאת הרשומה Microsoft Active Pages Server Object Library. סמן את תיבת הסימון בצד רשומה זו. כעת, גלול את הרשימה עוד מעט מטה, עד שתמצא את הרשומה Microsoft Transaction Server Type Library וסמן גם אותה. אם Active Server Pages ו-Microsoft Transaction Server מותקנים, יהיו שדות אלה זמינים ברשימה References. לחץ OK כדי לשמור בחירות חדשות אלו לפרויקט.

לאחר שספריות סוג אלו אוזכרו בפרויקט, ניתן ליצור מופעים של אובייקטים מתאימים. מופע האובייקט הראשון שעליך ליצור נקרא ObjectContext. זהו אובייקט Microsoft Transaction Server (MTS) המאחסן את המופעים הפעילים של כל האובייקטים המובנים של ASP, הקשורים ל-session מסוים. קיימת פונקציה גלובלית בשם ()GetObjectContext המחזירה ייחוס ל-ObjectContext זה.

האובייקט ObjectContext הוא אובייקט Collection, הדומה לאובייקטים Session ו-Application. הגישה לפריטים השונים באוסף נעשית באמצעות ערך מפתח הפריט. במקרה של אובייקטים מובנים של ASP, ערכי המפתח הם שמות האובייקטים (Request, Response, Server, Session ו-Application).

לכן, כדי לאחזר את האובייקט Response עליך להשתמש בקוד דומה לקוד הבא :

```
Dim objContext as ObjectContext
Dim objResponse as Response

Set objContext = GetObjectContext()
Set objResponse = objContext("Response")
```

# כתיבת קוד הרכיב

עתה, לאחר שנידונו היסודות הנחוצים ליצירת רכיבים מותאמים לשימוש בדפי ASP, הבט בפרטים נוספים. כמו כל רכיב COM, רכיבים מותאמים שנוצרו ב-Visual Basic יכולים להכיל מאפיינים (המייצגים את חלק הנתונים של האובייקט) ושיטות (המייצגות את חלק הפעולות של האובייקט).

Visual Basic מספקת מספר דרכים להוספת מאפיינים ושיטות למודולי מחלקה (זכור, מודול מחלקה הוא המקום בו מאחסנת Visual Basic קוד עבור אובייקט בודד בתוך הרכיב) :

- שימוש בתיבת הדו-שיח Add Procedure
- שימוש בשירות Class Builder
- הקלדה ידנית של שגרות המאפיינים או השיטות.

שתי השיטות הראשונות מתוארות בסעיף זה. לאחר שתדע כיצד נראית שיטה או מאפיין בקוד, רוב הסיכויים שתמצא את השימוש בשיטה השלישית לקלה ביותר.

## הוספת מאפיינים

מאפייני רכיב מייצגים את הנתונים המאוחסנים על ידי האובייקט. אתה יכול לחשוב על מאפיין כעל תכונה של אובייקט. לדוגמה, Color יכול להיות מאפיין לאובייקט Car.

מאפיינים יכולים להיות בעלי הרשאת Read/Write, Read-Only או Write-Only. משמעות הדבר היא, שבאפשרותך לשלוט על הגישה לנתוני האובייקט על ידי קביעת סכימת גישה מתאימה למאפיין.

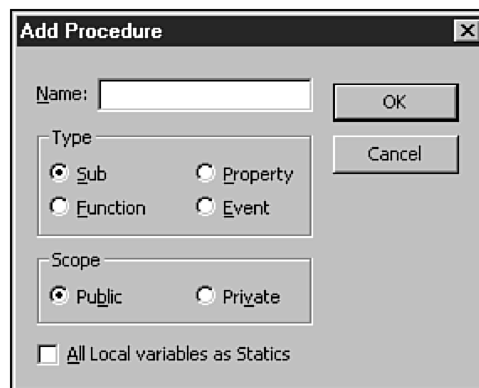
**הערה:**

אני ממליץ על השימוש במאפיינים בעלי הרשאת Write-Only. רק לעיתים נדירות תרצה למנוע מקוד לקוח את האפשרות לקרוא ערך של מאפיין. אחרי הכל, אם יש לך את הזכות לקבוע את ערך המאפיין, ודאי צריכה להיות לך גם הזכות לקרוא את אותו ערך בחזרה!

# הגדרת מאפיין באמצעות שימוש בפונקציות

במודול מחלקה של Visual Basic, מאפיינים יכולים להיות מיוצגים על ידי השגרות Property Get (לקריאת ערך המאפיין) ו-Property Let (לקביעת ערך המאפיין). אם ברצונך שמאפיין יהיה read-only, אל תכלול את השגרה Property Let. למאפיין write-only, אל תכלול את השגרה Property Get.

הוספת מאפיין הוא תהליך פשוט. פתח את חלון הקוד עבור מודול המחלקה של האובייקט שאליו ברצונך להוסיף את המאפיין (לחץ לחיצה כפולה על שם מודול המחלקה בסייר הפרויקט, או לחץ על שם מודול המחלקה ולאחר מכן על הלחצן View Code בסייר הפרויקט). לאחר שחלון הקוד הופך לפעיל, פתח את התפריט Tools ולחץ על פריט התפריט Add Property. תיבת הדו-שיח Add Procedure, כפי שנראית בתרשים 18.4 תופיע על המסך.



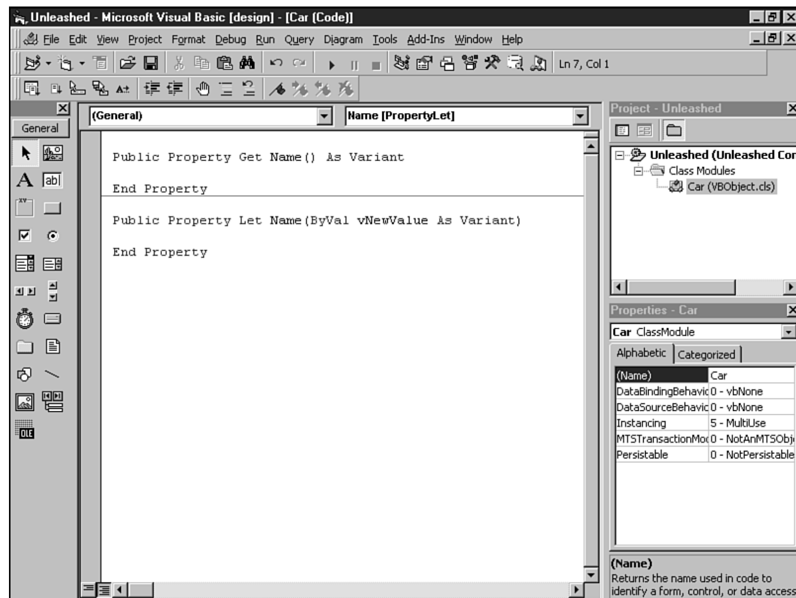

**תרשים 18.4** תיבת הדו-שיח Add Procedure.

בתיבת הטקסט Name, הקלד את שם המאפיין. אל תדאג לחלקים Property Get ו-Property Let - Visual Basic תוסיף אותם עבורך. הקלד רק את השם, כפי שאתה מעוניין שיופיע בקוד ה-ASP כאשר תשתמש בפרויקט למעשה.

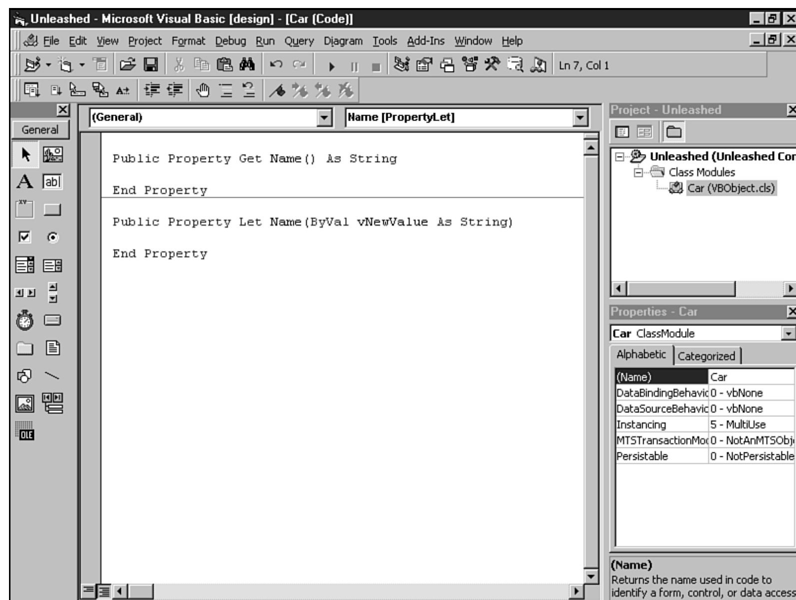
בקבוצת לחצני האפשרויות Type, בחר Property. השאר את לחצן האפשרויות Public מסומן, וקבע את תיבת הסימון All Local Variables as Statics בהתאמה (משתנה סטטי שומר על ערכו בכל פעם שמתבצעת קריאה לשגרה בה הוא נמצא. משתנים לא סטטיים מאופסים לערכי ברירת מחדל ידועים בכל פעם שמתבצעת קריאה לשגרה).

לחץ OK, וקטעי הקוד המתאימים יוספו למודול המחלקה. תרשים 18.5 מראה את חלון קוד מודול המחלקה לאחר שנוסף מאפיין בשם Name באמצעות תיבת הדו-שיח Add Procedure.

כיון שבתיבת הדו-שיח Add Procedure אין שום מקום בו ניתן לפרט את סוג נתוני המאפיין, מוסיפה Visual Basic את הקוד למאפיין מסוג Variant (אשר יכול להכיל כל סוג של נתונים, אולם יוצר תקורה רבה). כדי לקבוע את סוג הנתונים המתאים למאפיין, החלף את המילה Variant בשגרות Property Get ו-Property Let בסוג המתאים. לדוגמה, המאפיין Name הוא מחרוזת, ולכן בכל פעם שאתה רואה Variant החלף אותו ב-String (ראה תרשים 18.6).




**תרשים 18.5** מודול המחלקה לאחר הוספת המאפיין Name.




**תרשים 18.6** הקוד ששונה, מראה את סוג הנתונים המתאים עבור המאפיין Name.

כמו כן, תיבת הדו-שיח Add Property אינה מאפשרת לך לציין את סוג הגישה המורשית למאפיין (Read-Write, Read-Only, Write-Only). במקום זה, מכניסה Visual Basic את השגרות הן עבור קריאת ערך המאפיין והן עבור כתיבת ערך המאפיין. כדי להפוך את המאפיין ל-Read-Only, מחק את קטע הקוד של השגרה Property Let. למאפיין מסוג Write-Only, הסר את השגרה Property Get.

כעת, כל אשר נותר לעשות הוא לכתוב את הקוד עצמו עבור השגרות. ברוב המקרים, ערך המאפיין יאוכסן במשתנה מקומי המוגדר במודול המחלקה. במקרה של המאפיין Name, השתמשתי במשתנה מקומי בשם m_sName. הקידומת _m מציינת שזהו משתנה ברמת המודול. האות s מציינת שזו מחרוזת. החלק Name הוא תיאור הנתונים המאוחסנים במשתנה.

השגרה Property Get היא למעשה פונקציה של Visual Basic. כפי שניתן לראות בתרשים 18.6, לשגרה יש סוג נתונים שהוגדר לה: Property Get () As String. משמעות הדבר היא, שכדי להחזיר את ערך המאפיין, תצטרך לקבוע במקום כלשהו בשגרה את ערך ההחזרה לערך המאפיין. דבר זה נעשה ב-Visual Basic, כמו בדוגמה הבאה, על ידי קביעת שם הפונקציה לערך שאתה מעוניין שתחזיר הפונקציה (במקרה זה, זהו ערך המאפיין המאוחסן המשתנה המקומי m_sName):

```
Name = m_sNAme
```

עבור השגרה Property Let, הערך החדש המקושר למאפיין מועבר באמצעות השימוש בפרמטר האחד והיחיד של השגרה. תשתמש בערך זה כדי לקבוע את ערכו של המשתנה המקומי, בהנחה שהערך החדש יעובר את כל בדיקות התקינות של המאפיין.

תדפיס 18.1 מציג את הקוד המלא עבור המאפיין Name.

**תדפיס 18.1** השגרות Name Property Get ו-Name Property Let

```
Private m_sName as String

Public Property Get Name() As String

    Name = m_sName

End Property

Public Property Let Name(ByVal vNewValue As String)
' if empty string is passed' raise the
' invalid property value error (error # 380)
    If Len(vNewValue) = 0 Then Err.Raise 380

    m_sName = vNewValue

End Property
```

## שימוש במשתנה מקומי כבמאפיין

קיימת דרך קלה ליצור מאפיין באובייקט Visual Basic, אולם לקלות זו יש מחיר. אם תגדיר משתנה Public במקטע Declarations של מודול המחלקה, המשתנה ייחשף כמאפיין לכל לקוח COM המפעיל את האובייקט. להלן דוגמה:

```
Public Name as String
```

למרות שזה נראה נהדר, קיימים כאן מספר חסרונות:

- המאפיין הוא תמיד מאפיין מסוג Read/Write.
- אינך יכול לבצע שום בדיקת תקינות על הערך המוקצה למאפיין.
- הקוד אינו מקבל שום הודעה על שינוי בערכו של המשתנה המקומי. זאת אומרת, שאם לערך המאפיין יש השפעה על נתונים אחרים בתוך האובייקט, לא תוכל לשמור על הנתונים מעודכנים כאשר ישתנה ערך המאפיין.

## הוספת שיטות

הוספת שיטות (Methods) לאובייקט היא תהליך לוגי הדומה להוספת מאפיינים. אתה צריך להשתמש בתיבת הדו-שיח Add Procedure, ולבחור Sub או Function בקבוצת כפתורי הבחירה Type.

לפניך הדרך להוספת שיטה בשם Drive לאובייקט Car:

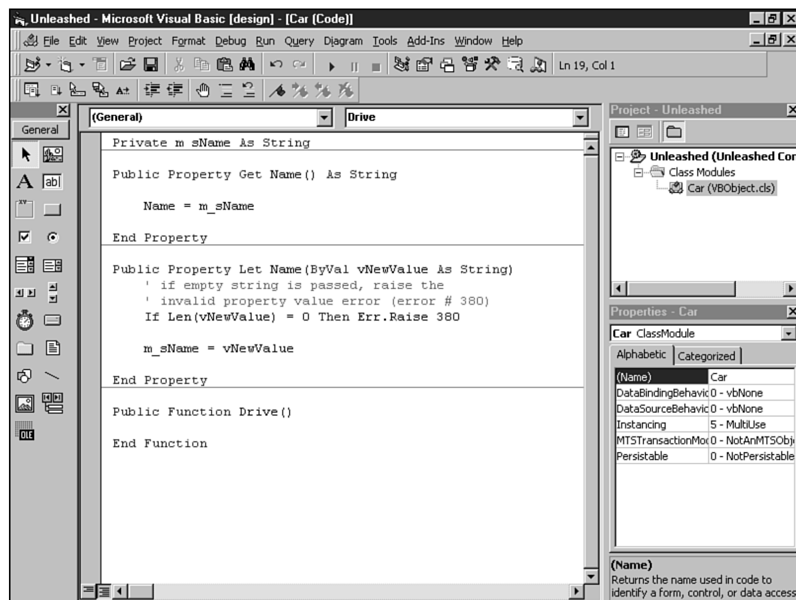
1. פתח את חלון הקוד למודול המחלקה של האובייקט Car.
2. פתח את תפריט Tools ובחר Add Procedure כדי להציג את תיבת הדו-שיח Add Procedure.
3. בתיבת הטקסט Name, הקלד את שם השיטה Drive. מאחר והשיטה תחזיר ערך שיציין אם המכונית החלה לנוע, בחר בלחצן האפשרויות Function. אם השיטה לא הייתה צריכה להחזיר ערך, היה עליך להשאיר את לחצן האפשרויות Sub הנבחר כברירת המחדל.
4. כיון ששיטה זו צריכה להיחשף על ידי האובייקט Car, השאר את לחצן האפשרויות Public נבחר. לחץ OK כדי להוסיף את הקוד למודול המחלקה. כעת, אמור המסך להיראות דומה למסך המוצג בתרשים 18.7 (הקוד עבור המאפיין Name נוסף בסעיף **הגדרת מאפיין באמצעות שימוש בפונקציות**, מוקדם יותר בפרק זה).
5. השיטה Drive תחזיר ערך בוליאני אשר יעיד על הצלחה או כשלון. כדי להוסיף קוד זה, שנה את השורה המגדירה את פרוצדורת השיטה לשורה זו:

```
Public Function Drive () As Bolean
```

**6.** כמו כן, השיטה חייבת לדעת כמה מהר לנסוע, והאם לנסוע קדימה או לאחור. נניח שהמהירות היא מספר שלם, ומשתמשים בפרמטר בוליאני (Boolean היכול לקבל ערך : True או False) אופציונלי אשר יציין נסיעה קדימה, שנה את אותה השורה לשורה הבאה :

```
Public Function Drive(Speed as Integer,
  ↳Optional Forward as Boolean = True) As Boolean
```

מילת המפתח Optional מציינת שקוד הלקוח אינו צריך להעביר את הפרמטר לשיטה. אם הלקוח אינו מעביר פרמטר זה, ייקבע למשתנה הפרמטר ערך ברירת מחדל (התלוי בסוג נתוני הפרמטר) או הערך שצוין, כאשר השגרה תבוצע. דבר זה נשלט על ידי מנוע ה-COM.



**תרשים 18.7** האובייקט Car לאחר הוספת השיטה Drive.

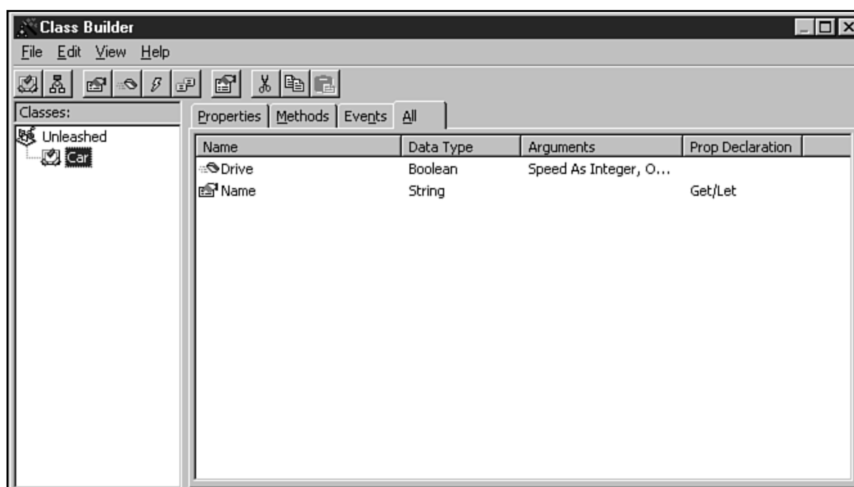
**7.** כעת, כל שנותר לעשות הוא לכתוב את הקוד אשר יגרום לאובייקט Car לנוע. תרגיל זה מושאר לך, הקורא!

# Class Builder Utility

Visual Basic מכילה תוכנית עזר נהדרת קטנה הנקראת Class Builder Utility. תוכנית זו מועילה ליצירה והוספה של אובייקטים לרכיב, ולהוספת קטעי השגרה למאפייני ושיטות הרכיב. אם הרכיב כולל מחלקת אוסף (Collection Class - מחלקה אשר כוללת אוסף של אובייקטים המוגדרים על ידי מחלקה אחרת ברכיב), ייצור Class Builder את רוב הקוד הדרוש כדי לגרום לזה לעבוד.

הבה ונסקור כמה מהתכונות של Class Builder. כיון ש-Class Builder היא תוכנית עזר, עליך לומר ל-Visual Basic לטעון אותו. השלבים המדויקים הדרושים לעשות זאת עברו שינוי מגירסה 5.0 לגירסה 6.0, אולם בשתי הגרסאות עליך להפעיל תחילה את מנהל תוכנית העזר (Add-In Manager) מתפריט Add-Ins. בשלב הבא, עליך לבחור בתוכנית העזר Class Builder ולאפשר אותה (גירסה 5.0) או לטעון אותה (גירסה 6.0). כמו כן, בגירסה 6.0 נוספו כמה תכונות של השירות, כך שאם אתה משתמש בגירסה 5.0 ייראה המסך שלך שונה במקצת. בכל אופן, הרעיון הבסיסי הוא זהה.

חלון Class Builder עבור האובייקט Car מוצג בתרשים 18.8. שים לב שהכרטיסיה All מונה הן את המאפיין Name והן את השיטה Drive.



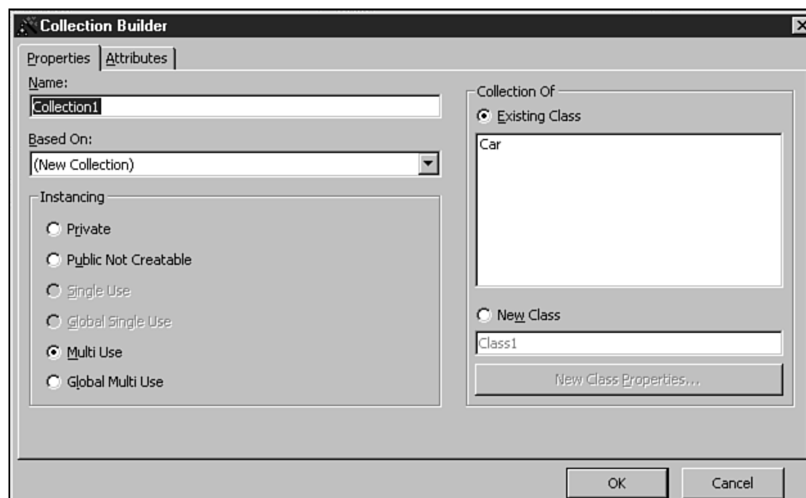

**תרשים 18.8** Class Builder Utility.

השימוש בשירות מאפשר לך לשנות שם, למחוק, לגזור, להעתיק, להדביק ולצפות במאפיינים של שיטות ומאפיינים קיימים. כמו כן, באפשרותך להוסיף מאפיינים חדשים, שיטות, אירועים או רשימות לאובייקט שלך.

### הערה:

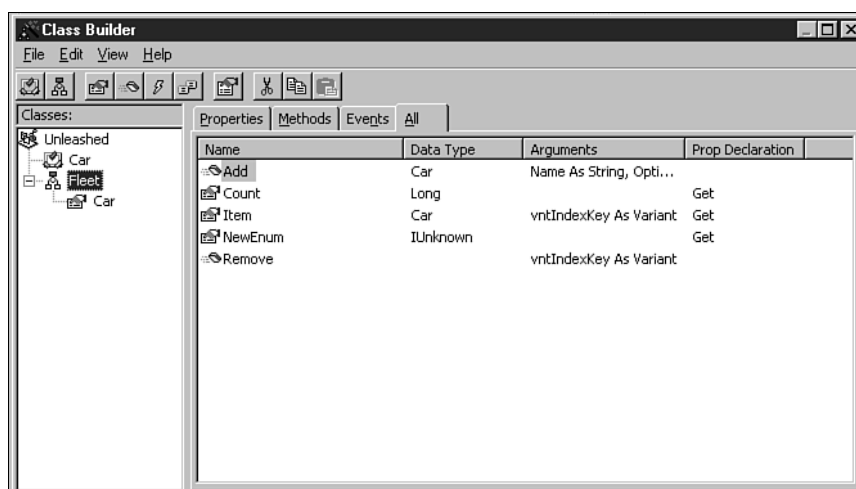
לא תוכל למחוק מחלקות שלא נוצרו באמצעות Class Builder. במקום זאת, מחק את מודול המחלקה מסייר הפרויקט של Visual Basic.

כדי להוסיף פרויקט חדש, לחץ על שם הפרויקט בתצוגת העץ, ולאחר מכן לחץ על Add New Class שבסרגל הכלים (זהו הלחצן השמאלי ביותר). כדי להוסיף מחלקת אוסף המכילה אוסף של אובייקטים המוגדרים ברכיב, עליך ליצור תחילה את המחלקה המגדירה את האובייקט להוספה. כעת, לחץ לחיצה ימנית על שם הפרויקט בתצוגת העץ, הצבע על New ומתפריט הקיצור בחר Collection. תיבת הדו-שיח Collection Builder (כפי שמוצגת בתרשים 18.9) תופיע.




**תרשים 18.9** תיבת הדו-שיח Collection Builder של Class Builder Utility.

ציין שם מתאים עבור המחלקה החדשה ובחר אילו מבין המחלקות הקיימות יכללו בה. במקרה זה, אני יוצר אובייקט בשם Fleet, אשר יהווה אוסף של אובייקטים מסוג Car. אני מקליד את השם Fleet בתיבת הטקסט Name ובוחר ברשומה Car מתיבת הרשימה Existing Class. לחיצה על OK, ו-Collection Builder תוסיף את מחלקת האוסף החדשה לתיבת הדו-שיח Class Builder, כפי שמוצגת בתרשים 18.10.




**תרשים 18.10** תיבת הדו-שיח Class Builder לאחר הוספת מחלקת האוסף.

כאשר תצא מהשירות Class Builder לאחר שביצעת שינויים כלשהם, תתבקש לעדכן את הפרויקט. אם ברצונך שהשינויים שביצעת אכן ישוקפו בפרויקט, הרשה ל-Class Builder לשנות את הקוד.

# השימוש ברכיב מדפי ASP

לאחר שמילאת את קוד הרכיב, עליך להדר אותו באמצעות הפקודה Make של Visual Basic (מתפריט File). פקודה זו תגרום להופעת תיבת הדו-שיח Make Project, בה ניתן לציין את שם קובץ ה-DLL של הרכיב, ואת התיקיה בה ייווצר. תוכל לקבוע אופציות הידור שונות, כגון מספר הגירסה עבור הרכיב ופרמטרים של הידור נוספים (אפשרויות אופטימיזציה, למשל). אפשרויות אלו זמינות כאשר תלחץ על Options בתיבת הדו-שיח Make Project.

לאחר הידור הרכיב, ניתן להשתמש בו בדף ASP. המהדר של Visual Basic דואג לכל דרישות ה-COM ברמה הבסיסית : יצירת מזהי CLSID ו-ProgID, רישום הרכיב וכדומה. אם תעביר את הרכיב למחשב אחר, יהיה עליך לרשום את הרכיב ברישום המערכת (Registry) של אותו מחשב (היעזר בפקודה REGSVR32.EXE כאשר שם קובץ ה-DLL הוא האופציה היחידה בשורת הפקודה), וכן ודא שכל התמיכה של זמן ריצה (Runtime) הדרושה זמינה.

כמו בכל רכיבי ActiveX/COM, באפשרותך ליצור מופע של רכיב Visual Basic בתוך דפי ה-ASP שלך, באמצעות השימוש בתגית <OBJECT> או השיטה ()Server.CreateObject.

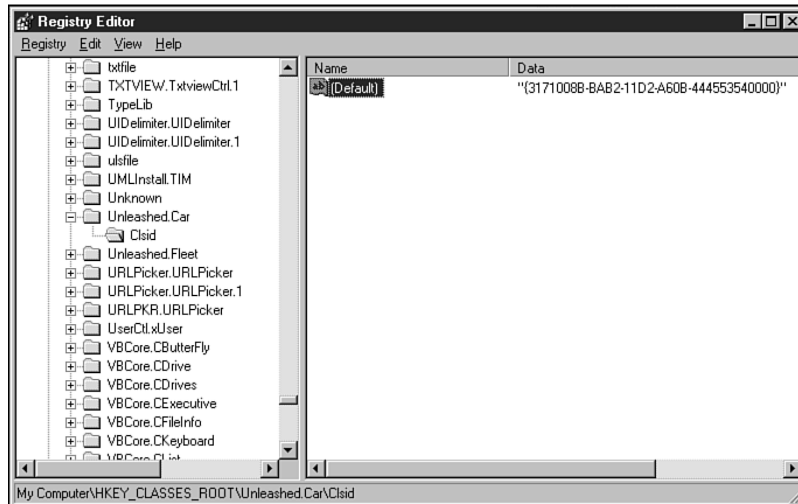
אם תשתמש בתגית <OBJECT>, יהיה עליך לציין את ה-CLSID עבור האובייקט. דבר זה ישפר את ביצועי דף ה-ASP שלך, כיון שאז לא יצטרך מנוע ה-ASP לבצע את הצעדים הבאים כדי לאתר את הרכיב. כיון ש-Visual Basic מייצרת את ה-GUID עבורך, אולם אין בסביבת הפיתוח מקום בו ניתן לראות את הערך שנוצר, עליך לחפש אותו ברישום המערכת של Windows, (ב-Registry). התחל את החיפוש במפתח HKEY_CLASSES_ROOT וחפש את המפתח המתאים עבור ה-PROGID של הרכיב. PROGID מבוסס על שם הפרויקט (Unleashed, כמו בדוגמה הקודמת), נקודה, ושם המחלקה (Car או Fleet, כמו בדוגמה הקודמת). כאשר תמצא את המפתח המתאים, הרחב את פריט העץ שלו, ותמצא מפתח בשם Clsid. ערך ברירת המחדל של מפתח זה הוא ה-CLSID של הרכיב. תרשים 18.11 מראה את הרשומה במחשב שלי עבור האובייקט Car. ה-CLSID הוא 3171008B-BAB2-11D2-A60B-444553540000, וה-PROGID הוא Unleashed.Car.

התחביר עבור התגית <OBJECT> הוא :

```
<OBJECT id="objID" classid="clsid:GUID"></OBJECT>
```

בתחביר זה, objID הוא השם בו תשתמש לאזכר את הרכיב בקוד תסריט דף ה-ASP שלך, והערך של GUID הוא ה-CLSID שמצאת ברישום המערכת. עבור האובייקט Car, צריכה התגית להיות :

```
<OBJECT id="myCar"
 classid="clsid: 3171008B-BAB2-11D2-A60B-444553540000">
</OBJECT>
```




**תרשים 18.11** רישום המערכת עבור האובייקט Car.

עבור השיטה CreateObject, הינך יודע כבר כל שאתה צריך. התחביר הוא:

```
Set myCar = Server.CreateObject("Unleashed.Car")
```

# היכרות עם היישום WebClass של Visual Basic

Visual Basic 6.0 מציגה סוג של יישום Visual Basic חדש לחלוטין: יישום IIS. יישומי IIS מוכרים בדרך כלל גם בשם יישומי WebClass, אולי משום ששם זה מצלצל נעים יותר לאוזן. יישום WebClass מאפשר לך להשתמש ב-Visual Basic כדי לכתוב קוד צד-שרת המגיב לבקשות מדפדפן המשתמש. בדיוק כמו בדפי ASP, הקוד יכול ליצור יישומים שאינם תלויי דפדפן.

ל-WebClass ישנה גישה לאובייקטים מובנים של ASP (Response, Request, Server, Session ו-Application). קוד WebClass אחראי להחזרת קוד ה-HTML להצגה, אולם משימה זו הופכת קלה באמצעות זמינות של אלמנטים מיוחדים של WebClass הנקראים פריטי תבנית HTML (HTML Template Items). פריטים אלה מכילים קוד HTML אשר יכול להיות מוצג כולו בו-זמנית, תוך שימוש בשיטה WriteTemplate.

כל יישום שאתה יוצר באמצעות ASP יכול להיות מיוצר גם באמצעות WebClass בסביבת הפיתוח של Visual Basic. הדבר כולל גם את היכולת לקבוע נקודות עצירה (Break Points), במיוחד עבור ניפוי שגיאות (Dedugging) בתוך קוד Visual Basic. יתרון נוסף של WebClass, אשר אינו זמין על ידי ASP, הוא חילוף תגיות עבור פריטי תבנית HTML. כפי שתראה, תבנית HTML יכולה להכיל תגיות מיוחדות אשר יגרמו להפעלת אירוע ProcessTag של פריט התבנית ביישום Visual Basic שלך. יתרון זה מאפשר לך להשתמש בפורמט סטנדרטי עבור טופס מסד נתונים (תבנית HTML) ועדיין לאפשר שינוי נתונים בצורה דינמית.

תכונה מלהיבה נוספת של WebClass היא היכולת ליצור ולהפעיל אירועים הנקשרים לפני/אחרי פעולה, ולעורר אירועים אלה ל-WebClass מהדפדפן. בהמשך הפרק נדון ביישומי IIS וביתרונות הרבים הטמונים בהם. הדוגמה המובאת אינה שימושית במיוחד, אולם היא מדגימה כיצד להשתמש בתכונות חשובות נוספות של Visual Basic WebClass.

**הערה:**

יישומי IIS יכולים לפעול על IIS בגרסאות 2.0, 3.0, או 4.0, Peer Web Services בסביבת Windows NT Workstations, או Personal Web Server בסביבת Windows 9x. לאובייקטים המובנים של ASP (Response, Request, Server, Session ו- Application) יש ערכות שונות של תמיכה במאפיינים ושיטות בין פלטפורמות שונות אלו. קח בחשבון אפשרות זו אם יישום ה-IIS שלך עתיד לעבוד בפלטפורמת שרת שונה מזו בה פותח היישום.

# יצירת יישום WebClass

בסעיף זה נדון בעקרונות הדרושים ליצירת יישום IIS חדש. עליך להתחשב בכמה דברים בתחילת תכנון היישום, כגון ניהול מצבים, ולאילו רכיבי שרת ActiveX של צד השרת תצטרך לקבל גישה.

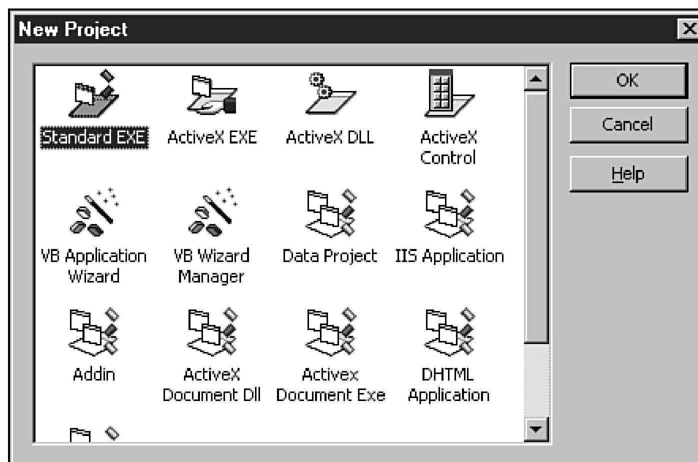
**הערה:**

מידע נוסף על יצירת יישום IIS חדש, וכן רשימת מאמרים הקשורים ל- WebClass תוכל למצוא בבסיס המידע של Microsoft, בכתובת

http://support.microsoft.com/support/kb/articles/q191/39.asp

## יצירת יישום IIS חדש

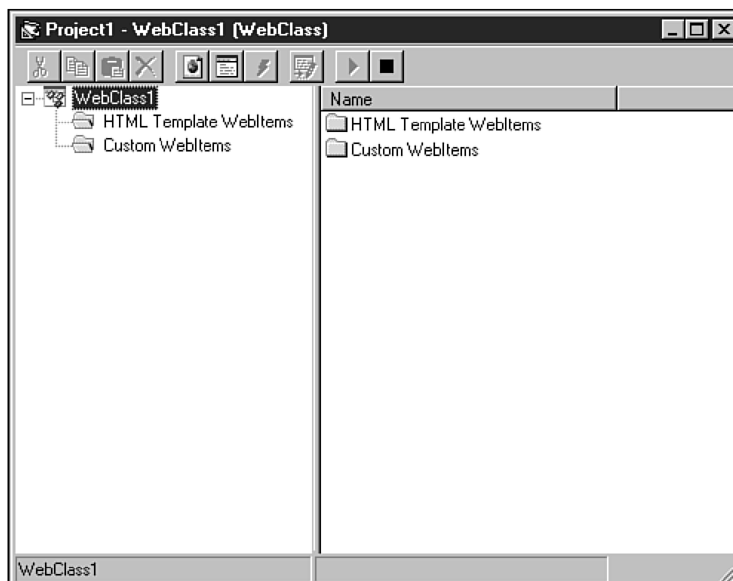
יצירת יישומי IIS/WebClass נעשית על פי אותם שלבים בהם יצרת יישום ActiveX DLL. אם הפעלת כרגע את Visual Basic מהתפריט **התחל** (Start) של Windows, תשתמש בכרטיסיה New של תיבת הדו-שיח New Project. אם Visual Basic כבר פועלת, פתח את תפריט File ובחר New Project, כדי להפעיל את תיבת הדו-שיח New Project (הנראית בתרשים 18.12).




**תרשים 18.12** תיבת הדו-שיח New Project

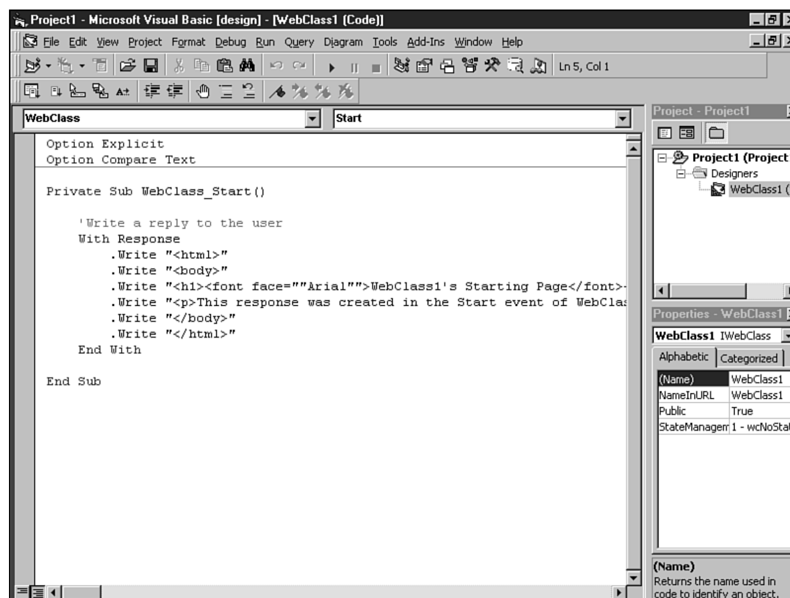
לחץ לחיצה כפולה על הסמל IIS Application ו-Visual Basic תיצור פרויקט חדש, שלם, ובו מספר קטעי קוד ראשוניים. שם הפרויקט יהיה כשם ברירת המחדל, Project1. כמו כן, נוצר אובייקט ברירת מחדל של WebClass, ושמו WebClass1. אובייקט זה הוא המקום בו תתבצע רוב עבודתך.

סביבת הפיתוח עבור WebClass מורכבת משני חלקים: מעצב WebClass (WebClass Designer המוצג בתרשים 18.13) וחלון הקוד. מעצב WebClass מחליף את מעצב הטפסים של Visual Basic, שבוודאי התרגלת אליו מיישומי Visual Basic סטנדרטיים. השימוש במעצב זה מאפשר לך להוסיף תבניות WebItems של HTML, להוסיף אובייקטי WebItems מותאמים, להוסיף אירועים לאובייקטי WebItems אלה, וכן להפעיל ולעצור את השרת המקומי.




**תרשים 18.13** מעצב WebClass

כאשר תעבוד עם מיגוון הרכיבים של יישומי IIS, תכיר את יתרונותיו של מעצב WebClass. כדי לראות את הקוד מאחורי פריט ב-WebClass, פשוט לחץ לחיצה כפולה על הפריט. לדוגמה, כדי לראות את הקוד הראשוני שהוגדר כברירת מחדל על ידי Visual Basic, לחץ לחיצה כפולה על הרשומה WebClass1 בחלון העץ של המעצב. יופיע חלון הקוד עבור WebClass (ראה תרשים 18.14).



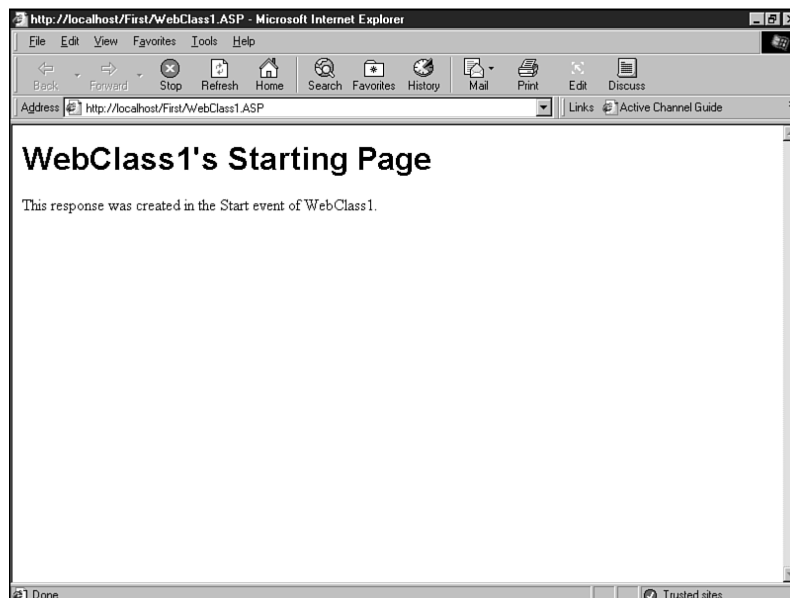

**תרשים 18.14** קוד ברירת המחדל של WebClass.

קוד ברירת המחדל שנוצר על ידי Visual Basic לפרויקט מספיק לשם הפעלת יישום IIS. קוד האירוע WebClass_Start מבוצע כאשר נוצר לראשונה אובייקט WebClass. כפי שניתן לראות בתרשים 18.14, קוד ברירת המחדל משתמש באובייקט Response כדי להציג קוד HTML פשוט לדפדפן.

לפני הפעלת הפרויקט החדש, עליך לשמור אותו. Visual Basic תיצור Web חדש בשרת האינטרנט המקומי, ותבקש ממך לבחור שם עבור Web זה (שם יישום ASP עבור התיקיה Web הווירטואלית שנוצרה). בנוסף, לפני הוספת תבנית ה-WebItem של HTML הראשונה לפרויקט, עליך לשמור אותו. תרשים 18.15 מראה את תוצאות הפעלת יישום IIS שנוצר כברירת המחדל. כאן קראתי ל-Web בשם First.

# גישה לאובייקטים מובנים של ASP

כפי שהצלחת אולי לפענח מהקוד המופיע בתרשים 18.14, יישום IIS מתחיל את חייו עם גישה לאובייקטים המובנים של ASP (Request, Response, Server, Session ו-Application). אינך צריך לעשות שום דבר מיוחד כדי להפוך אובייקטים אלה לזמינים כפי שעשית ביישומי ActiveX DLL סטנדרטיים. תמיכה באובייקטים אלה מהודרת לתוך יישום ה-DLL על ידי המהדר של Visual Basic.




**תרשים 18.15** הפלט המוצג על ידי יישום IIS הנוצר כברירת המחדל.

כפי שניתן לראות בקוד ברירת המחדל, אלה הם אותם אובייקטים אליהם התרגלת מכתיבת תסריטי ASP. הצגת HTML לדפדפן נעשית באמצעות השיטה Response.Write(), בדיוק כמו בכתיבת קוד ל-ASP. ויש גם יתרון נוסף: האובייקט BrowserType, שיתואר מאוחר יותר בפרק זה, בסעיף **הגדרת מאפיין באמצעות שימוש בפונקציות**

# גישה לרכיבי שרת נוספים של ActiveX

כמו בכל הסוגים של יישומי Visual Basic, יישומי IIS יכולים להפעיל רכיבי ActiveX אחרים. לדוגמה, אם בידך קוד שמבצע פעולות כלשהם המובנה ברכיב ActiveX, באפשרותך לגשת לפונקציונליות זו מ-WebClass בפרויקט שלך. פונקציונליות זו יכולה לכלול הכל, החל מאימות כניסה למערכת (Login Verification) ועד לעיצוב נתונים מותאם אישית.

לא משנה מהי הפונקציונליות, השימוש בה בתוך WebClass הוא פשוט. ראשית, עליך ליצור ייחוס לרכיב החיצוני בתוך הפרויקט. דבר זה מאפשר לפרויקט להשתמש בקריאות הנקשרות לפני פעולה למאפייני ושיטות האובייקט. השימוש בקריאות הנקשרות לפני פעולה יעיל הרבה יותר.

כדי להוסיף ייחוס לרכיב חיצוני, הפעל את תיבת הדו-שיח References של Visual Basic (פתח את תפריט Project ובחר References). תיבת הדו-שיח References מאפשרת לך לבחור כל אחד מרכיבי השרת של ActiveX הרשומים כרגע במחשב, או לבצע רישום של רכיב חדש, אם יש ברשותך את קובץ ההפעלה שלו (DLL או EXE). כדי לבחור רכיב שכבר רשום במערכת, אתר את הרשומה שלו בתיבת הרשימה Available References, וסמן את תיבת הסימון לידה. כדי לרשום רכיב חדש למערכת, או כדי לאתר רכיב על פי שם הקובץ שלו, השתמש בלחצן Browse (**עיון**). לחצן זה יציג את תיבת הדו-שיח הסטנדרטית File Open (**פתיחת קובץ**), המאפשרת לך לאתר ולבחור את קובץ ההפעלה של הרכיב.

לאחר שיצרת ייחוס לרכיב, תוכל להשתמש בו באמצעות יצירת משתנה מופע (Instance Variable), כדי שיחזיק במופעי האובייקט המסופקים על ידי הרכיב. לדוגמה, הקוד הבא יוצר מרכיב בשם Unleashed משתנה חדש בשם objMyFleet מהסוג Fleet, וכמו כן מפעיל מופע פעיל של האובייקט:

```
Dim objMyFleet as New Unleashed.Fleet
```

לאחר שהשלמת פעולה זו, ניתן להשתמש בכל השיטות והמאפיינים המסופקים על ידי האובייקט.

# שמירת מצב האובייקט

מושג חשוב נוסף שעליך להתחשב בו בעת תחילת הפיתוח של יישום ה-IIS שלך הוא שמירת מצב. כלומר, שתוכל לשלוט על האופן בו יישמר מצב ה-WebClass מבקשה לבקשה. ליישום IIS הנוצר ב-Visual Basic יש שני אופנים אפשריים של שמירת מצב: שמור או לא שמור. הערך מצוין באמצעות שימוש במאפיין StateManagement של האובייקט WebClass. למאפיין זה יש שני ערכים: wcNoState או wcMaintainInstance.

אם תבחר לשמור את מצב ה-WebClass מבקשה לבקשה, תשתמש בזיכרון נוסף בשרת (כיון שה-WebClass נשמר "בחיים" באובייקט Session לכל אורך session המשתמש), אולם תוכל לשמור על המשתנים (המקומיים) הפנימיים של WebClass מבקשה לבקשה. שים לב ששימוש באובייקטים מסוג Session דורש מהדפדפן לתמוך ב-cookies, לכן ודא שהמשתמשים ערים לצורך זה. אם תבחר שלא לשמור על המצב, ישוחרר הזיכרון המשמש את WebClass מייד לכשתושלם התגובה, אך לא תוכל לשלוט בערכי המשתנים הפנימיים על פני בקשות רבות.

פעולתם של משתני האובייקט Session אינה מושפעת כלל מקביעה של המאפיין StateManagement. תמיד תוכל לבחור בערך wcNoState ולהשתמש במשתני Session כדי לאחסן את הנתונים המשמשים על פני בקשות רבות (בדיוק כפי שעשית עם קוד ASP סטנדרטי). בכל אופן, דרך זו היא בניגוד למרקם הסטנדרטי של רוב הפיתוח ב-Visual Basic.

אם תבחר בערך wxMaintainInstance ותרצה להיפטר מהמופע הנוכחי, כדי לשחרר זיכרון או כדי לסיים את השימוש בו, השיטה ()ReleaseInstance תמלא בדיוק מטלה זו. היישום יסתיים בסיום השגרה בה הופעלה השיטה ()ReleaseInstance.

# עבודה עם WebItems

עתה, לאחר שדנו ביסודות יישומי IIS, זה הזמן לבדוק מה דרוש כדי שיישום יעשה עבודה רצינית יותר. למרות שהאירוע WebClass_Start מסוגל לטפל בבקשה יחידה באופן מושלם, כדי לבנות יישום שימושי עליך להוסיף עוד כמה אלמנטים. ביישומי IIS נקראים אלמנטים אלה WebItems.

WebItems משומשים על ידי WebClass כדי לספק תוכן ואירועים לדפדפן. כל WebClass יכול להכיל מספר רב של WebItems. WebItems מאוחסנים בתוך WebClass, ולא כמודולים נפרדים בתוך הפרויקט. את ה-WebItems שתוסיף ל-WebClass תמצא בחלונית העץ של חלון מעצב WebClass.

ישנם שני סוגים של WebItems: תבנית HTML WebItem, ו-Custom WebItem. כדי להוסיף WebItem ל-WebClass, פתח את מעצב WebClass ולחץ על לחצן סרגל הכלים Add HTML Template WebItem, או על לחצן סרגל הכלים Add Custom WebItem. אם אתה מוסיף תבנית HTML WebItem, תופיע תיבת הדו-שיח File Open. השתמש בתיבת דו-שיח זו כדי לאתר את קובץ ה-HTML שישמש כתבנית ל-WebItem החדש.

## שימוש בתבנית HTML WebItem

תבנית HTML WebItem מאפשרת לך לקשר קובץ HTML ל-WebItem. לפני שתוסיף WebItem ל-WebClass, אתה חייב ליצור את קובץ ה-HTML. באפשרותך להשתמש בכל עורך HTML (כולל Notepad) כדי ליצור את הקובץ. קוד HTML הנכלל בקובץ יכול להיות מסמך HTML מלא, או רק חלק ממסמך HTML שיוכנס לתוך מסמך HTML אחר.

כאשר תוסיף תבנית HTML WebItem ל-WebClass ותשמור את הפרויקט, תיצור Visual Basic עותק של קובץ התבנית. אם הקובץ המקורי לא אותר בספריית הפרויקט, הוא פשוט ישוכפל לספריית הפרויקט באותו שם. אם הקובץ המקורי נמצא בספריית הפרויקט, תיצור Visual Basic עותק של הקובץ, ותוסיף לשם הקובץ המקורי ספרה. לדוגמה, השם template.htm יכול להשתנות ל-template1.htm. מנקודה זו עליך לעבוד עם הקובץ המשוכפל, כדי לשנות את הופעת הפלט.

כדי להציג את תכולת התבנית של מחרוזת התגובה עליך להשתמש בשיטה ()WriteTemplate של WebItem. ניתן להוסיף קריאות לשיטה ()WriteTemplate בכל מקום בו אתה יוצר תגובה. תכולת קובץ התבנית תתווסף לתגובה.

ניתן גם להעביר את שטף מחרוזת התגובה ל-WebItem מסוים, על ידי קביעת המאפיין הגלובלי NextItem של WebClass ל-WebItem זה. לאחר שמסתיימת השגרה הנוכחית, יופעל אירוע Respond המסוים של WebItem זה, המאפשר לו לכתוב את תוכן מחרוזת התגובה לדפדפן. הקוד הבא ממחיש רעיון זה:

```
Private Sub WebClass_Start()
    with Response
        .Write "<HTML>"
        .Write "<BODY>"
        .Write "<H1><FONT face=" "Arial" ">"
        .Write "WebClass1's Starting Page</FONT></H1>"
        .Write "<P>This response was created in the "
        .Write "start event of WebClass1.</P>"
    End With
    Set NextItem=Piece     'transfer control to Piece
End Sub

Private Sub Piece_Respond()
    Response.Write "In the Piece_Respond event: "
    Piece.WriteTemplate
End Sub
```

יתרון נוסף, ויעיל במיוחד, של תבנית HTML WebItem הוא האפשרויות שלו להכיל חילוף תגיות אשר יגרמו להפעלת האירוע ProcessTag של WebItem, כאשר תופעל השיטה ()WriteTemplate. דבר זה מאפשר לך להתאים אישית את תכולת מחרוזת התגובה המהווה חלק מקוד ה-HTML שמכילה התבנית. בנושא חילוף התגיות נדון בסעיף **תכונות מיוחדות של יישומי WebClass**, מאוחר יותר בפרק זה.

# הוספת WebItem מותאם אישית

לאלמנטי WebItem מותאמים אישית אין קובץ HTML המקושר אליהם. WebItem מכיל אירועים (Event Handlers) אשר מופעלות כאשר מופעל WebItem (על ידי קביעת המאפיין NextItem) או כאשר מתרחשת פעולה מסוימת כלשהי בדפדפן. ניתן להוסיף ל-WebItem אירועים בשעת התכנון, או להשתמש באירוע UserEvent כדי לייצר אירועים דינמיים (פריטים נוספים בסעיף הבא).

כיון של-WebItem מותאם אישית אין קובץ תבנית, השיטה ()WriteTemplate אינה תקפה. במקומה, השתמש באירוע Respond, או באירועים אחרים בצורה תכניתית, כדי לייצר את מחרוזת התגובה.

# תגובה לאירועים

רוב העבודה הנעשית על ידי יישום IIS נעשה בתוך ה-Event Handler (אירועים). אירועים מופעלים כתגובה לפעולות שנעשו בדפדפן, או כאשר אלמנטים מיוחדים בתוך מחרוזת התגובה מעובדים. אנו נדון בארבעה סוגי אירועים : אירועים מובנים (Build-In Events), אירועי תבנית (Template Events), אירועים מותאמים (Custom Events), והאירוע UserEvent.

אירועים קיימים באובייקטים של WebClass ובשני הסוגים של WebItems. ערכת האירועים הסטנדרטית לתבניות HTML ול-Custom WebItems, זהה.

## אירועים מובנים

ישנם כמה אירועים מובנים עבור WebClass ועבור ה-WebItems המרכיבים אותו. אירועים אלה מופעלים כתגובה לאירועים ברמת מערכת או כפעולת ברירת מחדל כאשר אין שום אירוע אחר לטפל בבקשה מהדף הנוכחי ביישום ה-WebClass. אינך צריך לעשות כלום כדי לגשת לחלון הקוד עבור אירועים אלה. הם הוספו אוטומטית על ידי Visual Basic. האירועים המובנים מתומצתים בטבלה 18.1.

**טבלה 18.1** האירועים המובנים של יישום IIS

| אובייקט | אירוע |
|---|---|
| WebClass | Start |
| | BeginRequest |
| | EndRequest |
| | Initialize |
| | Terminate |
| | FatalErrorResponse |
| WebItem | Respond |
| | UserEvent |
| | ProcessTag |

את האירוע הראשון כבר ראית. האירוע Start מופעל בכל פעם בו מבוקש הדף הראשי של יישום WebClass. דבר זה נעשה כאשר נעשות בקשות עבור ProjectName.ASP, כאשר ProjectName הוא המאפיין NameInURL של WebClass (אשר שווה לשם ה-WebClass כברירת מחדל), או כאשר מבוקש WebItem אשר אינו קיים.

האירועים BeginRequest ו-EndRequest מופעלים בכל פעם בו מתבצעת בקשה כלשהי ליישום. דבר זה כולל בקשות עבור WebItems, בנוסף ל-WebClass עצמו. אם אתה צריך לבצע עיבוד מרוכז לכל דף המבוקש ביישום, אלה האירועים בהם יש להשתמש.

האירוע BeginRequest מופעל בתחילת הבקשה, לפני הפעלת כל אירוע אחר. האירוע EndRequest מופעל לאחר שהסתיימו כל האירועים האחרים (כולל האירוע ProcessTag). דבר זה מתרחש מייד לפני ששרת האינטרנט מחזיר את תוכן מחרוזת התגובה לדפדפן.

האירועים Initialize ו-Terminate זהים לאירועים Initialize ו-Terminate המוכרים למתכנתי רכיבי ActiveX. אירועים אלה מופעלים בכל פעם בו נוצר או נהרס מופע של האובייקט, בהתאמה.

האירוע FatalErrorResponse מופעל בכל פעם בו מופסק תהליך העיבוד של WebClass עקב טעות. לשגרת אירוע זה יש פרמטר בודד, SendDefault as Boolean, אשר משמש לומר לשרת האינטרנט האם האירוע יצר את הודעת השגיאה המתאימה.

**אזהרה:**

אם אתה משתמש בשיטה Err.Raise של Visual Basic כדי להפעיל שגיאות בתוך אובייקט WebClass, האירוע FatalErrorResponse לא יופעל אלא אם כן משהו כבר נכתב למחרוזת התגובה.

אם הפרמטר SendDefault נקבע בקוד ל-False, אזי עליך ליצור את הודעת השגיאה להחזרה (באמצעות השימוש בשיטה Response.Write או בשיטה WriteTemplate של תבנית HTML WebItem). ל-WebClass יש מאפיין בשם Error אשר כולל את מספר השגיאה, תיאור ומקור. ניתן להשתמש במאפיין זה כדי לקבוע איזו שגיאה התרחשה, ולעדכן בהתאם את הדפדפן. שים לב שכל הנתונים אשר כבר נפלטו למחרוזת התגובה (על ידי השימוש בשיטה Response.Write או השיטה WriteTemplate) עדיין יוצגו כאשר יופעל אירוע השגיאה. באפשרותך להשתמש בשיטה Response.Clear כדי להסיר את הנתונים הקיימים. אם תשאיר את ערך הפרמטר SendDefault כ-True, הדבר יגרום לשליחת הודעת ברירת המחדל של שגיאת ASP לדפדפן.

עבור WebItems, האירוע Respond מופעל בכל פעם בו מתבצעת בקשה ל-WebItem, ואין אירוע עבור האלמנט הנבחר בדף הנוכחי. אירוע זה הוא אירוע ברירת המחדל עבור WebItem, ותגובת הדפדפן חייבת להיווצר בתוך האירוע. אחרת, היישום לא יפעל. בשני האירועים הנוספים של WebItem מטבלה 18.1 (UserEvent ו-ProcessTag) נדון בסעיפים הבאים.

## קישור אירועים מותאמים אישית לפני פעולה

בזמן התכנון, באפשרותך ליצור אירועים משלך אשר יופעלו מהדפדפן. ניתן ליצור אירועים בתבנית HTML WebItem או ב-Custom WebItem. בדרך כלל יצירתם תהיה ב-Custom WebItem.

אירועים מותאמים אישית מופעלים על ידי לחיצה על קישורי HyperText בתוך דפים שנוצרו על ידי אירועים או WebItems אחרים. הספריה WebClass מספקת שיטה שימושית גלובלית בשם URLFor, בה ניתן להשתמש לבניית ה-URL עבור WebItem והאירועים שלו. בדוגמת יישום ה-IIS בהמשך תוכל לראות כיצד דבר זה פועל.

כדי ליצור אירוע חדש, פתח את חלון מעצב WebClass. לחץ על ה-WebItem אליו ברצונך להוסיף את האירוע החדש (לא ניתן להוסיף אירועים ל-WebClass עצמו, אלא רק ל-WebItems שלו), ולחץ על הלחצן Add Custom Event שבסרגל הכלים (נמצא בחלון מעצב WebClass, לא בסרגל הכלים של Visual Basic). ענף חדש המייצג אירוע חדש זה נוסף לעץ ה-WebItems. כדי לשנות את שם האירוע, לחץ לחיצה ימנית בעץ ומתפריט הקיצור בחר Rename. כדי לכתוב קוד עבור אירוע, לחץ לחיצה כפולה על האירוע בעץ, או פתח את חלון הקוד של WebClass, והשתמש בתיבת הרשימה הנגללת בתחילת חלון הקוד כדי לאתר את האירוע.

כדי ליצור קישור טקסט שיפעיל אירוע מסוים, השתמש בקוד דומה לקוד הבא:

```
Response.Write "<A HREF="
Response.Write URLFor(WebItem, "EventName")
Response.Write ">Invoke Event</A>"
```

בכל פעם שקישור הטקסט יילחץ בדפדפן, ינסה WebClass להפעיל אירוע בשם EventName ב-WebItem המצוין על ידי WebItem (החייב להתייחס לאובייקט מסוג WebItem). שים לב ששם האירוע מסומן במרכאות, כלומר שעליך להעביר מחרוזת עבור פרמטר זה. אם לא קיים אירוע כזה ב-WebItem, יופעל האירוע UserEvent של WebItem (ראה בסעיף הבא). אם תציין WebItem שאינו קיים, תייצר השיטה URLFor הודעת שגיאה בזמן ריצה.

## האירוע UserEvent

בכל פעם בו מאוזכרת כתובת URL ומצויים בה אירוע לא ידוע, נקרא האירוע UserEvent של WebItem. דבר זה שימושי כדי ליצור אירועים דינמיים שאינם ידועים כאירועים של זמן-תכנון, אולי כיון שהם מבוססים על נתונים הנשלטים על ידי WebClass. יצירת קישור ל-UserEvent נעשית בדיוק כפי שנעשית יצירת קישור לאירוע הנקשר לפני פעולה:

```
Response.Write "<A HREF="
Response.Write URLFor(WebItem, "EventName")
Response.Write ">Invoke Event</A>"
```

EventName הוא השם שיועבר לאירוע UserEvent. קטע הקוד עבור האירוע UserEvent הוא:

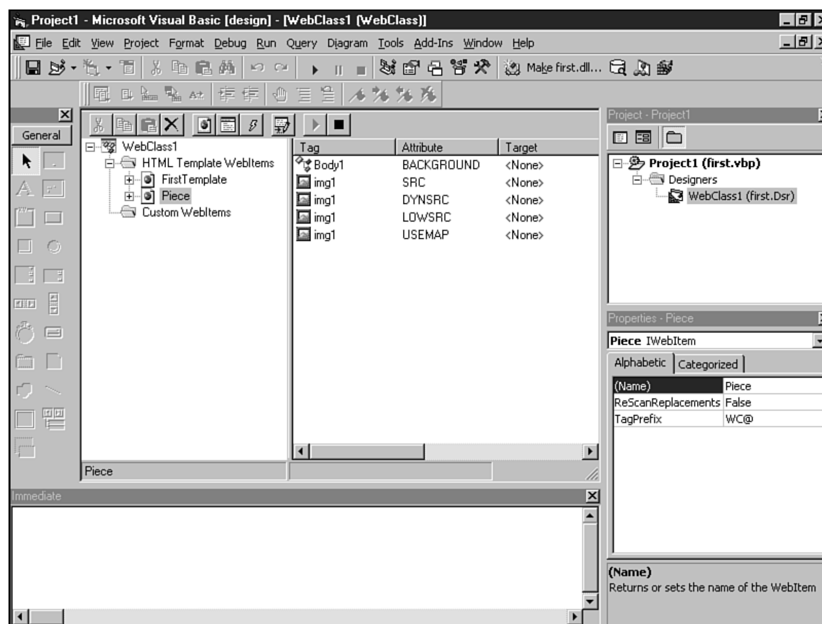
```
Private Sub WebItemName_UserEvent(ByVal EventName As String)
```

הפרמטר EventName המועבר לאירוע זהה לערך שסופק בקישור.

## אירועי פריט תבנית

האלמנטים של WebItems מציגים סוג מיוחד נוסף של אירוע: אירועי פריט תבנית (Template Item Events). ניתן לשלוט במאפיינים מסוימים של תגיות HTML מסוימות על ידי שימוש באירועי פריט תבנית. לדוגמה, אם קיימת תגית <IMG> בתוך קובץ תבנית, ניתן ליצור בצורה דינמית את התכונות SRC, DYNSRC, LOWSRC ו-USEMAP.

בכל פעם שאתה מוסיף תבנית HTML WebItem ל-WebClass אשר כוללת תגיות עם תכונות מתאימות עבור אירועי פריט תבנית (אלמנטי טופס, תגיות תמונה, קישורי טקסט, ותגיות רבות אחרות המכילות אזכור של כתובת URL), התגיות והתכונות מוספות לרשימת ה-WebItem של מעצב WebClass (ראה תרשים 18.16).




**תרשים 18.16** WebClass Designer עם תבנית HTML המכילה תגיות מתאימות עבור Template Item Events.

כדי להוסיף אירוע לתכונה מתאימה כלשהי, לחץ לחיצה כפולה על הרשומה המתאימה לתכונה בחלון הרשימה, ו-Visual Basic תפתח את חלון הקוד עבור אירוע זה. השתמש בשיטות ()Response.Write ו-()Response.BinaryWrite כדי לספק את הפלט הדרוש.

שים לב שהאירוע אינו מצפה ממך לספק את כתובת ה-URL עבור התכונה של התגית. במקום זאת, עליך להחזיר את המידע אותו כתובת ה-URL מצפה לתאר. לדוגמה, עבור התכונה SRC של התגית <IMG>, מצפה ממך האירוע לכתוב את נתוני התמונה למחרוזת הפלט:

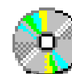
```
Private Sub Piece_img1Src()
    Response.BinaryWrite LoadResData(101, "custom")
End Sub
```

קוד זה יקרא את קובץ המשאב המקושר לפרויקט ה-WebClass, ויכתוב את הנתונים למחרוזת הפלט.

# טיפול בקלט מטופס

כאשר אתה משתמש ביישום WebClass כדי לטפל בטפסי HTML, אין באמת הבדל רב מכתיבת תסריטי ASP (לפחות בכל הקשור לקוד עצמו). נתוני הטופס מוכלים באוסף Request.QueryString (אם התכונה METHOD של הטופס היא GET) או באוסף Request.Form (אם התכונה METHOD של הטופס היא POST).

בתגית <FORM> של טופס HTML, קבע את התכונה ACTION לאירוע באחד מה-WebItems של ה-WebClass, על ידי כתיבת קוד ה-URL (אם אתה משתמש בתבנית HTML), או על ידי שימוש בשיטה URLFor (אם אתה משתמש ב-Response.Write כדי להציג את הטופס), או על ידי חילוף תגיות HTML המתואר בהמשך הפרק (אם אתה משתמש בתבנית HTML וברצונך להשתמש בשיטה URLFor כדי ליצור את ה-URL המתאים). לחילופין, תוכל פשוט להשתמש ב-Custom WebItem ולעבד את נתוני הטופס באירוע Respond של WebItem. באירוע, גש לאוסף המתאים עבור שיטת הטופס הספציפית (GET או POST).



שילוב של עיבוד נתוני הטופס עם קטעי HTML ו-Custom Template WebItem אחרים עושה את השליטה בזרימת היישום על בסיס נתוני הטופס לפשוטה מאוד. לדוגמה, אולי תהיה מעוניין לנתב משתמשים לדף אחר של טופס רישום בהסתמך על המידע שמילאו בטופס. תוך שימוש בשיטה Response.Redirect, או אם יש לך WebItems עבור דפים אחרים, באפשרותך לבצע משימה זו ללא כל בעיה, על ידי קביעת המאפיין NextItem. תדפיס 18.2 מציג דוגמה פשוטה מאוד לביצוע משימה זו (הפרויקט כולו נמצא בתקליטור, בתיקיה Chapter18\forms).

```
Private Sub WebClass_Start()

    'Write a reply to the user
    With Response
        .Write "<html>"
        .Write "<body>"
        .Write "<h2><font face=""Arial"">WebClass Form Processing</font></h2>"
        .Write "<form method=post action=" & URLFor(FormSubmit) & ">"
        .Write "Name: <input type=text name=username><p>"
        .Write "Marital Status: <select name=mstatus>"
        .Write "<option value=0 SELECTED>Select status..."
        .Write "<option value=1>Married<option value=2>Single<option
                 value=3>Other</select><p>"
        .Write "<input type=submit></form>"
        .Write "</body>"
        .Write "</html>"
    End With

End Sub

Private Sub FormSubmit_Respond()

    With Response
        .Write "Hello " & Request.Form("username") & "<br>"

        Select Case Request.Form("mstatus")
        Case 0
            Response.Redirect ("myformclass.asp")
        Case 1
            Set NextItem = WI_Married
        Case 2
            Set NextItem = WI_Single
        Case 3
            Set NextItem = WI_Other
        Case Else
            Response.Redirect ("myformclass.asp")
        End Select

    End With

End Sub

Private Sub WI_Married_Respond()

    Response.Write "Welcome to the Married Folks Forum!"

End Sub
```

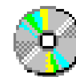
```
Private Sub WI_Other_Respond()

   Response.Write "Welcome to the Other Folks Forum!"

End Sub

Private Sub WI_Single_Respond()

   Response.Write "Welcome to the Single Folks Forum!"

End Sub
```

כדי ליצור פרויקט זה מהתחלה, צור יישום IIS חדש. שנה את המאפיינים Name ו-NameInURL של WebClass ל-MyFormClass. בשלב הבא, הוסף את ה-WebItems המותאמים אישית: FormSubmit, WI_Married, WI_Single ו-WI_Other. כעת העתק את הקוד המופיע בתדפיס 18.2 והפעל את היישום.

# תכונות מיוחדות של יישומי WebClass

כעת, לאחר שהכרת את יסודותיהם של יישומי WebClass, זה הזמן לבחון מספר תכונות מתקדמות זמינות. סעיף זה דן במאפיין URLData, חילוף תגיות תבנית HTML, והאובייקט BrowserType.

## נתונים מתמידים עם המאפיין URLData



בנוסף לשימוש במשתנה האובייקט Session לשמירה על נתונים, ניתן לשמור את מצב היישום שלך על ידי שימוש במאפיין URLData. מאפיין זה, אשר מוכר ל-WebClass בצורה גלובלית, הוא מחרוזת המצורפת ל-URL של כל WebItem ולכל אירוע של היישום. כעת, תוכל להשתמש בשיטה Request.QueryString, או במאפיין URLData, כדי לקבוע את המצב הנוכחי. כאשר אתה משתמש במאפיין QueryString, המפתח בתוך האוסף יהיה WCU. דוגמת יישום ה-WebClass (נמצאת בתקליטור, בשם UnleashedWCProject.vbp) משתמשת במאפיין URLData.

## שימוש בחילוף תגיות של תבניות HTML

תכונה נוספת ליצירת דפי אינטרנט דינמיים היא חילוף התגיות (Tag Replacement), הזמינה כאשר משתמשים בקבצי תבניות HTML. על ידי הוספת תגיות מיוחדות לקובץ ה-HTML, ניתן לגרום להפעלת האירוע ProcessTag של WebItem. דבר זה מאפשר לך להחליף את תוכן התגית בתוכן דינמי. האירוע מוגדר כ:

```
Private  Sub TemplateItem_ProcessTag(ByVal TagName As String,
   ➥TagContents As String, SendTags As Boolean)
```

הפרמטר TagName מציין את שם התגית המיועדת להחלפה. לכל פריט תבנית HTML יש מאפיין בשם TagPrefix, המציין את המחרוזת המשמשת לזיהוי התגית המחליפה. ערך ברירת המחדל היא @WC. תגית ה-HTML, בהנחה של ברירת מחדל זו, בה תשתמש בכל פעם שתרצה להפעיל את האירוע ProcessTag היא:

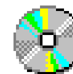
<WC@TagName>TagContents</WC@TagName>

הנתונים הממוקמים בתוך תוחמי התגית (מיוצגים בדוגמה זו על ידי TagContent) מועברים לאירוע בתוך הפרמטר TagContent. פרמטר זה הוא גם פרמטר פלט: אירוע קובע את ערכו של פרמטר זה לנתונים בהם תשתמש. נתונים אלה יכולים להיות טקסט פשוט או HTML.

הפרמטר SendTags משמש כדי לציין אם התגיות שגרמו לאירוע ProcessTag גם מוחזרות לדפדפן. ערך ברירת המחדל הוא False, מה שאומר שרק ערכו של הפרמטר TagContent יישלח לדפדפן. אם ערך הפרמטר SendTags נקבע ל-True, יישלחו לדפדפן גם התגיות <WC@TagName> ו-</WC@TagName>.

# שימוש באובייקט BrowserType

התכונה המיוחדת האחרונה הזמינה ביישומי WebClass היא האובייקט BrowserType. אובייקט זה זהה לאובייקט שנדון בפרק 7, אולם הוא זמין בצורה גלובלית ב-WebClass, ללא כל צורך בקוד.



תדפיס 18.3 מראה כיצד ניתן להשתמש באובייקט BrowserType כדי להעביר שליטה ל-WebItem אחר, בהסתמך על דפדפן המשתמש. (קוד זה נמצא בתקליטור, בתיקיה BrowserType). ל-WebClass זה מוספים שלושה איברי WebItem מותאמים אישית: IEPages, NetscapePages ו-UnknownPages.

**תדפיס 18.3** עיבוד נתוני טופס

```
Private Sub WebClass_Start()

  If InStr(BrowserType.Browser, "IE") Then
    Set NextItem = IEPages
  ElseIf InStr(BrowserType.Browser, "Netscape") Then
    Set NextItem = NetscapePages
  Else
    Set NextItem = UnknownPages
  End If

End Sub

Private Sub IEPages_Respond()

  With Response
    .Write "You're using version " & BrowserType.version
    .Write " of Internet Explorer"
  End With

End Sub
```

```
Private Sub NetscapePages_Respond()

    With Response
        .Write "You're using version " & BrowserType.version
        .Write " of Netscape"
    End With

End Sub

Private Sub UnknownPages_Respond()

    With Response
        .Write "You're using version " & BrowserType.version
        .Write " of " & BrowserType.Browser
    End With

End Sub
```

# סיכום

פרק זה כיסה חומר רב. למדת כיצד להשתמש ב-Visual Basic (מגירסה 4.0 ומעלה) כדי ליצור רכיבי COM הנגישים מתוך דפי ASP שלך. כמו כן, למדת אודות מחלקת יישום חדשה, שימושית אף יותר, אשר הוצגה לראשונה ב-Visual Basic גירסה 6.0 : יישום IIS. באמצעות שימוש ביישום IIS באפשרותך להחליף לחלוטין את כל דפי ASP שלך ביישום Visual Basic בודד.

פרק 19 מתאר כיצד להשתמש ב- ++Visual C ובספריית תבניות פעילות (ATL - Active Template Libraries) כדי ליצור רכיבי COM קטנים, מהירים, הנגישים מתוך דפי ASP.

## פרק 19

# יצירת רכיבים באמצעות Visual C++

**בפרק זה:**

- יתרונות השימוש ב-Visual C++
- Threading Models
- יצירת הרכיב (Component)
- דוגמה פשוטה

שני הפרקים הקודמים עסקו ביצירת רכיבי צד-שרת שמשתמשים בתסריטים וב-Visual Basic. פרק זה מדגים כיצד לבנות רכיב כזה תוך שימוש בסביבת הפיתוח של Visual C++. ספציפית, הרכיב ייבנה תוך שימוש ב-Active Template Library (ATL) של Microsoft. למידע נוסף אודות Visual C++ ו-ATL, פנה לאתר MSDN של Microsoft בכתובת http://msdn.microsoft.com.

ATL היא הדרך הקלה והמהירה ביותר לבניית רכיבי COM בעזרת Visual C++. אשפי ATL מקלים רבות על מלאכת הוספת מחלקות COM (COM Classes), השיטות והמאפיינים שלהן. אם אתה חייב להשתמש ב-MFC בפרויקטים שלך, ATL מספקת תמיכה ב-MFC, ואני אראה לך כיצד להוסיף תמיכה זו לרכיב שלך.

פרק זה מכסה כמה נושאים :

- יתרונות השימוש ב-Visual C++ לבניית רכיבי צד-שרת
- באילו מודלי מטלות עליך להשתמש עבור הרכיבים שלך
- כיצד ליצור רכיב פשוט.

# יתרונות השימוש ב-Visual C++

אם קראת את שני הפרקים הקודמים, אתה מודע ללא ספק לכך שתהליך יצירת הרכיבים בשימוש בתסריטים או ב-Visual Basic קל מאוד. כל אחת משתי השיטות הללו יכולה להפיק רכיב בזמן מרשים למדי, ותוך דרישה לכתיבת כמות קטנה מאוד של קוד מצידך. כפי שתראה בפרק זה, פיתוח רכיב COM ב-Visual C++ יצריך יותר זמן קידוד מצידך. אם כך, מדוע Visual C++ כל כך מושכת?

להלן כמה יתרונות בשימוש ב-Visual C++ לפיתוח רכיבי צד-שרת :

- ביצועים - ללא ספק, Visual C++ מפיקה את הרכיבים המהירים ביותר, בהשוואה לרכיבים שנוצרו על ידי תסריטים (אותם יש לפרש במלואם בזמן אמת) ו-Visual Basic (רכיבים מהודרים, אך עדיין עוברים פירוש חלקי על ידי מערכת "Virtual Machine" של VB). בנוסף, השימוש בתסריטים דורש זמן נוסף לניתוח וביצוע הקוד. רכיבי C++ מהודרים, ולכן הם פועלים בצורה טבעית.
- העלאת רמת הקוד הקיים - אם יש לך כבר לוגיקה עסקית שכתובה ב-Visual C++, פשוט מאוד להעביר לוגיקה זו לרכיב צד-שרת.
- הגנה - שמור על הקוד שלך מפני עיניים בולשניות. אם אתה מפיץ את הרכיבים שלך לחברות אחרות, סביר להניח שתרצה להגן על זכויות היוצרים שלך, בכל הנוגע לקניין רוחני. לא ניתן לעשות זאת על ידי שימוש בתסריטים. Visual Basic מספקת קצת יותר הגנת קוד, אך עדיין ניתן להשתמש במהדר הפוך (DeCompiler) כדי לפענח את הקוד.

כשאתה משתמש ב-ATL, אתה משיג כמה יתרונות נוספים:

- רכיבים קטנים ומהירים יותר
- תמיכה בכל מודלי המטלות (למרות שרק שניים מהם מומלצים עבור רכיבי צד-שרת)
- הוספת תמיכה בממשקי COM נוספים שהרכיב שלך מטמיע, היא כמעט טריוויאלית
- תוכל לעשות שימוש פנימי בכמות עצומה של מחלקות נוספות (כגון CComBSTR ו-CComPtr)
- תמיכה במנגנון טיפול השגיאות של COM.

# Threading Models

אם, כמוני, יש לך רקע בפיתוח תוכנה עבור מחשב בודד, סביר להניח שלא התרגלת לכך שייתכן ויהיה צורך ברכיב כדי לשרת כמה לקוחות בו-זמנית. משמעות הדבר היא שבכל זמן נתון הרכיב יכול להפעיל שגרה אחת עבור לקוח אחד ושגרה אחרת לגמרי עבור לקוח אחר.

מערכת ההפעלה Windows מטפלת במציאות זו על ידי כך שהיא מאפשרת לרכיב לפעול בריבוי מטלות (Multiple Threads) בו-זמנית, כלומר, קוד הרכיב יכול להתבצע ביותר מנתיב אחד בזמן נתון. Windows מספקת לכל מטלה (thread) פרוסת זמן ספציפית לפעולה, ולאחר מכן עוברת לבצע את המטלה האחרת. הרכיב שלך ינצל את תהליך העיבוד מרובה המטלות (multithreaded) על פי מודל המטלות (threading model) בו תבחר כשתיצור את הרכיב. ישנם ארבעה מודלי מטלות שונים: Single, Apartment, Free ו-Both.

**הערה:**

למידע נוסף אודות מטלות ומודלי מטלות, פנה לאתר Microsoft בכתובת האינטרנט http://msdn.microsoft.com/msdn-online/workshop/components/com/comthread.asp.

כשאתה יוצר רכיב, אתה יכול להשתמש בכל אחד מהמודלים האלה. עם זאת, רק מודלי apartment ו-both יספקו ביצועים סבירים בסביבת Active Server Page מרובת משתמשים.

## רכיבי Single-Threaded

מודל Single-Threaded הוא המודל הגרוע ביותר מתוך קבוצת מודלי המטלות, כשמדובר ברכיבי צד-שרת. כשרכיב מטלה Single-Threaded פועל, כל קריאה לאחת מהשיטות שלו מבוצעת במטלה יחידה. משמעות הדבר היא שלאחר שמתבצעת קריאה, שאר הקריאות שעשויות להתבצע ייחסמו ולא ניתן יהיה להפעיל אותן עד לאחר שהקריאה הקודמת הסתיימה. גרוע מכך, אם כמה משתמשים מנסים לגשת לרכיב בו-זמנית, עלול לקרות מצב נעילה (Deadlock).

אם זה לא מספיק כדי להבריח אותך מרכיבי Single-Threaded, עליך להיות מודע גם לכך שרכיבים אלה פועלים תוך שימוש בחשבון ה-SYSTEM, כלומר שלרכיב יהיו את כל ההרשאות הקשורות בחשבון SYSTEM. בדרך כלל, לא תרצה לפתוח את המערכת שלך לגישת דפדפני אינטרנט.

## רכיבי Free-Threaded

מודל מטלות Free-Threaded אינו מומלץ גם הוא לפיתוח רכיבי צד-שרת. ראשית, יש לרשום את רכיבי Free-Threaded ב-Microsoft Transaction Server (MTS) כדי לגשת לאובייקט ObjectContext (בו משתמשים לצורך תמיכה בהעברות בתוך רכיב או ב-Active Script Page). רכיבי Apartment-Threaded ו-Both-Threaded יכולים לגשת לאובייקט ObjectContext ללא רישום ב-MTS.

בנוסף, רכיבי Free-Threaded, כמו רכיבי Single-Threaded, פועלים תוך שימוש בחשבון SYSTEM. דבר זה כמובן שאינו רצוי כי אנחנו רוצים שכל מי שנכנס דרך הרשת יוכל להיכנס למערכת.

## רכיבי Apartment-Threaded

ניתן להשתמש ברכיבי Apartment-Threaded ברכיבים שיש להם טווח של דף או Session יחיד (כלומר, השימוש ברכיבים הוא בדף ASP יחיד או שהם מוצבים במשתנה Session). עם זאת, כשאתה יוצר רכיב Apartment-Threaded ברמת ה-Session, ה-Session כולו נעול במטלה יחידה. דבר זה מאט את זמן הגישה. אך אם הרכיבים שלך פועלים רק במשך זמן החיים של דף יחיד, רכיבים אלה יפעלו כשורה.

אם אתה מנסה להקצות רכיב Apartment-Threaded למשתנה Application, תקבל שגיאת הפעלה (Runtime Error). במילים אחרות, אם הרכיב Unleashed.ApartmentThreaded הוא מסוג Apartment-Threaded, הקוד הבא יגרום לשגיאה:

```
Set Application ("Oops") = Server.CreateObject("Unleashed.ApartmentThreaded")
```

## רכיבי Both-Threaded

ניתן להשתמש במודל רכיבי Both-Threaded ברכיבי צד-שרת, ללא קשר לטווח המיועד לו. אך למען היעילות, יש ליצור את הרכיב כאובייקט Marshaler מסוג Free-Threaded. באובייקט זה משתמשים כדי לבצע קריאות בין המטלות בלי צורך ב-Marshaling או ב-Threaded Switches. אשף ATL New Object מקל על הוספה של תמיכה כזו למחלקות שלך (כפי שתראה כשתיצור רכיב בהמשך הפרק).

אם אתה יוצר רכיב Both-Threaded ואינך יוצר אובייקט Marshaler מסוג Free-Threaded, תוכל להשתמש ברכיב בטווח יישום רק אם פרמטר התצורה MD_ASP_TRACK_THREADING_MODEL ייקבע כ-True (1). ברירת המחדל של פרמטר זה היא False (0). אחרת, תתקבל הודעת שגיאת הפעלה כשתנסה להקצות מופע (instance) של הרכיב למשתנה בטווח יישום. ברמת ה-Session, לא תקרה שגיאת הפעלה, אך אם לא תיצור את אובייקט ה-Marshaler מסוג Free-Threaded, תשפיע לרעה על הביצועים.

החסרונות בשימוש ברכיבי Both-Threaded הוא שאינך יכול לשרשר קריאות לרכיב, לפיכך עליך ליצור אבטחת מטלות בקוד שלך. למרות שצורה זו תאריך את זמן הפיתוח, ייתכן והיא תהיה נחוצה עבור רכיבי טווח Session או טווח יישום.

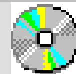
# יצירת הרכיב

כעת, לאחר שאתה כבר יודע באיזה מודל מטלות לבחור (Both-Threaded אם הרכיב שלך צריך לתמוך בטווח כלשהו ו- Apartment-Threaded אם הרכיב בטווח דף), ראה כיצד הרכיב נוצר תוך שימוש ב-Visual C++ 6.0 ו-ATL 3.0.

ישנם שלושה שלבים כלליים הדרושים ליצירת רכיב צד-שרת בעזרת Visual C++ ו-ATL:

1. צור פרויקט ATL COM חדש.
2. הוסף אובייקט ATL פשוט לכל מחלקה שהרכיב שלך יספק.
3. הוסף את המאפיינים והשיטות הנתמכים על ידי מחלקות הרכיב.

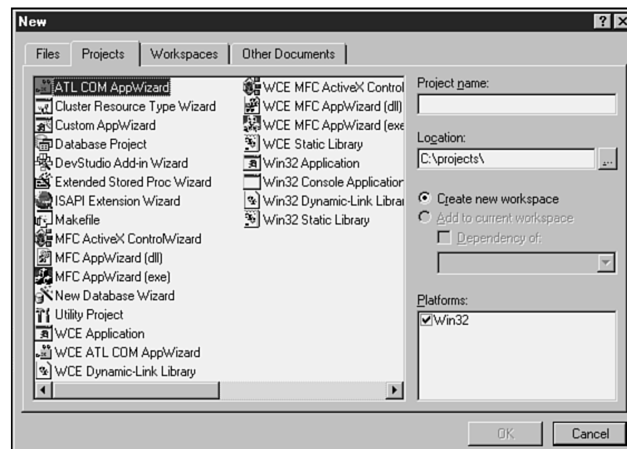
**הערה:**



תוכל למצוא את הגירסה הסופית של רכיב זה בתקליטור המצורף, בתיקיה Chapter19\first

# יצירת פרויקט ATL

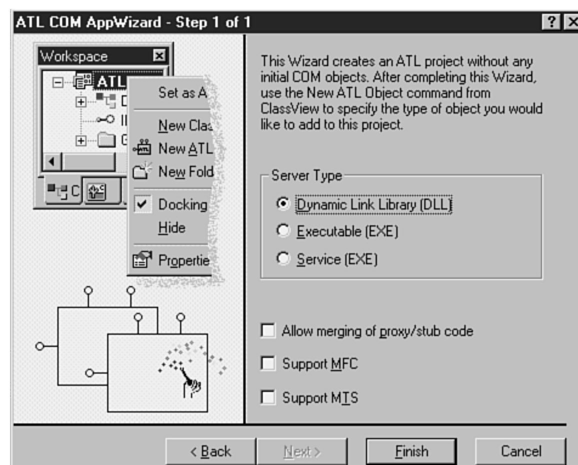
קל ליצור פרויקט חדש. הפעל את Visual C++ מתוך תפריט "התחל" של Windows. בשלב זה, תקבל שטח עבודה ריק. בחר באפשרות New מתוך תפריט File. כעת תופיע תיבת הדו-שיח New המוצגת בתרשים 19.1.




**תרשים 19.1** תיבת הדו-שיח New של Visual C++.

ראשית, ודא שהכרטיסיה Projects נבחרה. בתצוגת הרשימה שבצד השמאלי של תיבת הדו-שיח, לחץ על ATL COM AppWizard. הקלד את שם הפרויקט בתיבת הטקסט Project Name, ודא שהתיבה Location מתאימה. האשף ייצור תיקיית משנה ששמה הוא שילוב שם המיקום ושם הפרויקט (תבין למה אני מתכוון בזמן שתקליד את הנתונים בתיבת הטקסט Project Name).

לאחר שהקלדת את הנתונים, לחץ OK והאשף יופעל, כפי שמוצג בתרשים 19.2.




**תרשים 19.2** תיבת הדו-שיח היחידה של אשף ATL COM AppWizard.

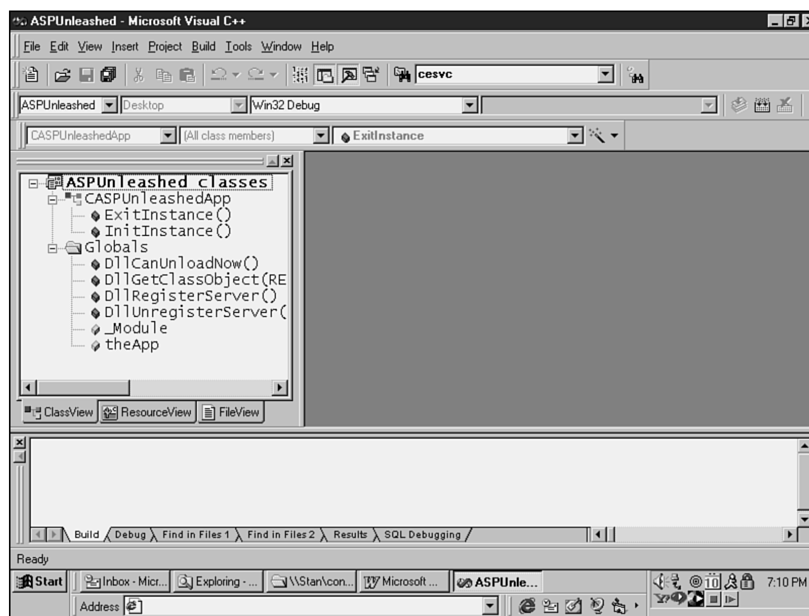
כללית, רכיבי צד-שרת יוטמעו כ-DLL (משום ש-DLL פועל תוך כדי התהליך, ולפיכך ביצועי הרכיבים טובים יותר). רכיבי DLL הם ספריות אשר פועלות בתוך אותו מרחב הכתובות שניתן לתוכנית ממערכת ההפעלה. לכן, רכיב DLL יפעל מהר יותר מרכיב exe שרץ בתהליך משלו ובמרחב כתובות משלו. השאר את אפשרות Server Type בערך ברירת המחדל שלה. אם, מסיבה מסוימת, תזדקק לרכיב שפועל מחוץ לתהליך, בחר באפשרות Executable (EXE). שים לב שייתכן ויהיה עליך לשנות גם כמה אפשרויות בשרת IIS כדי להתאים אותו לרכיבי out of process (למידע נוסף, פנה לכתובת http://msdn.microsoft.com/msdn-online/workshop/server/components/outproc.asp)

אם תשתמש בכל אחת ממחלקות MFC (וישנן כמה מחלקות שימושיות שאינן UI ב-MFC) או אם אתה סבור שתזדקק להן אי פעם, סמן את תיבת הסימון Support MFC. תוכל לחזור לכאן מאוחר יותר כדי להוסיף תמיכת MFC ברכיב, אך יהיה עליך להוסיף כמות מסוימת של קוד לשם כך (פנה למאמר Microsoft Knowledge Base מספר Q173974, למידע אודות הוספת תמיכת MFC לפרויקט ATL).

אם הרכיב שלך יירשם ב-MTS, סמן את התיבה Support MTS.

לאחר שסיימת, לחץ על לחצן Finish והאשף יציג תיבת דו-שיח שמתארת את הפרויקט שאתה מתכוון ליצור. לחץ OK בדו-שיח זה, והאשף ייצור את שלד הפרויקט החדש שלך.

לאחר סיום הפעולה, תופיע סביבת הפיתוח כפי שמוצג בתרשים 19.3 (לאחר הרחבת עץ שטח העבודה).




**תרשים 19.3** תוצאת פעולת אשף ATL COM AppWizard.

מה יש בידך בשלב זה? אשף ATL COM AppWizard יצר את מעטפת הרכיב. מעטפת זו מורכבת מהקוד הנדרש להידור ולטעינת ה-DLL כשרת COM. השרת לא עושה עדיין כלום, משום שהוא אינו מכיל מחלקות COM (הידועות גם בשם CoClasses). מה שיש לך כרגע הוא הצד השמאלי של פרמטר ProgID, בו אתה משתמש כשאתה קורא לשיטה Server.CreateObject() (ברכיב שמוצג בתרשים 19.3, פרמטר זה יהיה ASPUnleashedApp).

**הערה:**

המילה **שרת**, המוזכרת פה, מתייחסת לרכיב הפועל בתהליך ומשרת את היישום ולא למחשב מרוחק.

# הוספת מחלקה חדשה

עליך להוסיף CoClass כדי להשלים את הרכיב שלך. ה-CoClass הוא המקום בו יוטמעו שיטות ומאפייני הרכיב. בעולם ה-ATL, דבר זה נעשה על ידי בחירת אפשרות New ATL Object מתוך תפריט Insert בעורך Visual C++. לאחר שבחרת בפריט התפריט הזה, יופעל אשף ATL Object Wizard, כפי שמוצג בתרשים 19.4.

**תרשים 19.4** אשף ATL Object Wizard.

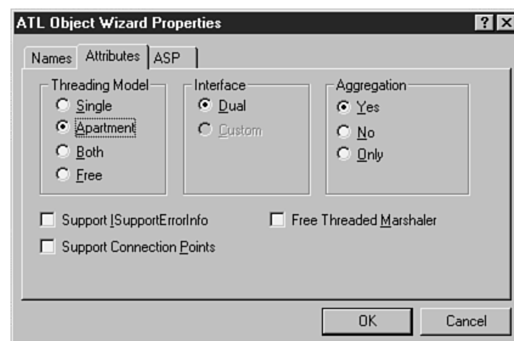
תיבת דו-שיח זו מאפשרת לך להוסיף סוגים שונים של אובייקטי ATL לפרויקט שלך. הקטגוריה Objects מכילה פריטים שמוטמעים כאובייקטי COM, כגון MMC Snapin. תחת הקטגוריה Controls תמצא פקדי ActiveX רבים שניתן ליצור אותם תוך שימוש ב-ATL. הקטגוריה Miscellaneous מכילה מחלקה אחת, Dialog. הקטגוריה Data Access מכילה מחלקות שמיועדות ליצירת ספקי וצרכני OLE DB.

אתה יוצר רכיב שמיועד לשימוש בדפי ASP, לכן תרצה ליצור רכיב ActiveX Server, שנמצא בקטגוריה Objects. לחץ על רכיב ActiveX Server Component ולאחר מכן לחץ על לחצן Next. כעת תוצג תיבת הדו-שיח ATL Objects Wizard Properties, כמוצג בתרשים 19.5.

**תרשים 19.5** תיבת הדו-שיח ATL Object Properties.

עבור רכיב שרת ActiveX, יש לתיבת דו-שיח זו שלוש כרטיסיות: Names, Attributes, ו-ASP. הכרטיסיה Names היא המקום בו אתה מציין את שם המחלקה החדשה שאתה יוצר. הערך שיופיע בסופו של דבר בתיבה CoClass ישמש כחלק הימני של ProgID בו משתמשים כדי לקרוא ל- ()Server.CreateObject. תוכל להקליד ערך זה בתיבת הטקסט Short Name, והאשף ימלא את כל התיבות האחרות עבורך. לרוב, תשאיר את התיבות האחרות עם הערכים שהתקבלו על ידי האשף, אך תוכל לעקוף אותן. כרטיסיה זו מאפשרת לך גם להשתמש בשמות שונים עבור מחלקות ++C (שנמצאות בשימוש פנימי על ידי הקוד שלך) וממשק COM CoClass (בו משתמש לקוח ה-COM חיצונית, כגון דף ASP).

לאחר שאתה מרוצה מהערכים, בחר בכרטיסיה Attributes, המוצגת בתרשים 19.6. בכרטיסיה זו אתה מציין את התכונות השונות של CoClass. תכונות אלו כוללות את מודל המטלה (threading model), המתואר קודם לכן בפרק זה), שיטת הממשק (למרות שרכיבי ActiveX Server יכולים להיות רק בעלי ממשק כפול), ואם ניתן לצבור (Aggregate) את האובייקט. בחירתך במודל המטלה מוכתבת על ידי טווח הרכיב בתוך יישום ה-ASP שלך, אך כדי לקבל גמישות מירבית בחר באפשרות Both.



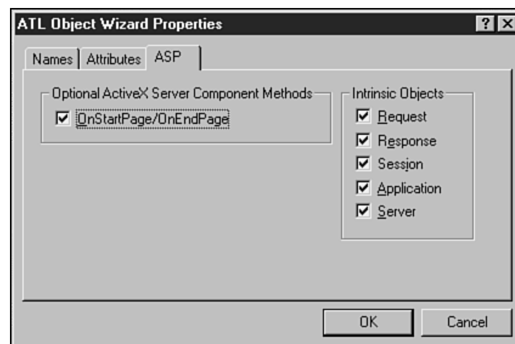

**תרשים 19.6** הכרטיסיה Attributes של תיבת הדו-שיח ATL Object Wizard Properties

אם תרצה לנפות שגיאות מסוג COM מהאובייקט שלך, סמן את תיבת Support IsupportErrorInfo. דבר זה יגרום להוספת הקוד הדרוש לשימוש במנגנון טיפול שגיאות ה-COM ברכיב שלך.

אתה משתמש במודל Both, ולכן תרצה שיהיה גם אובייקט Marshaler מסוג Free בסביבה. לפיכך, סמן את התיבה הנקראת Free-Threaded Marshaler כדי שהאשף יוסיף את כל הקוד הנחוץ ל-Marshaler מסוג Free.

השתמש בתיבת הסימון Support Connection Points אם הרכיב שלך יעביר אירועים למארח (Host) שלו. דף ASP אינו יכול להגיב לאירועים של רכיבי COM שהוא יוצר, ולכן ברוב המקרים אין צורך לסמן תיבה זו.

לבסוף, הכרטיסיה ASP (מוצגת בתרשים 19.7) מכילה כמה תיבות סימון שהאשף משתמש בהן, כדי ליצור את אובייקטי ה-ASP הפנימיים הזמינים לאובייקט שלך. כדי לאחזר את המופעים הפעילים של אובייקטים פנימיים אלה, אתה חייב להטמיע את השיטה OnPageStart. אתה יכול לבטל סימון כל אחד מהאובייקטים שהרכיב שלך אינו צריך לגשת אליו. בכך תסיר את תקורת האחזור והשחרור של מופעי אובייקטים אלה.



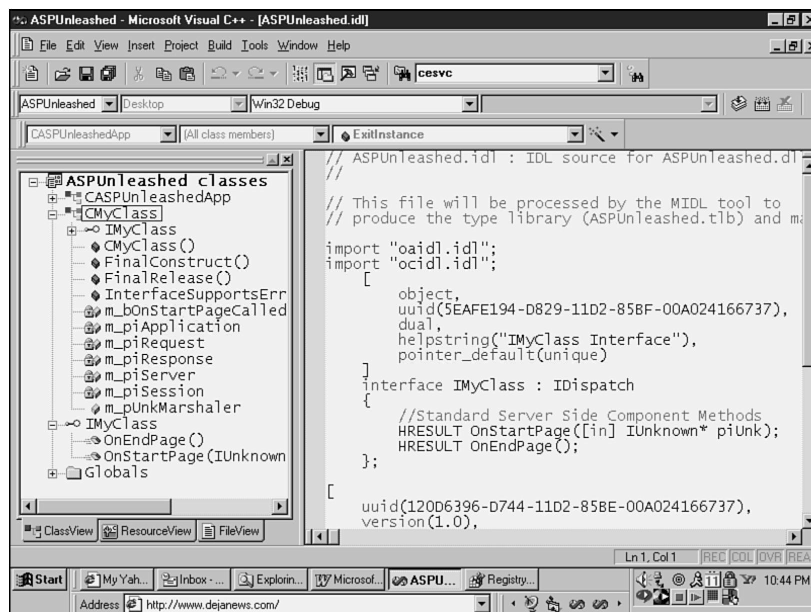

**תרשים 19.7** הכרטיסיה ASP של תיבת הדו-שיח ATL Object Wizard.

לאחר שמצב תיבות הסימון משביע רצון, לחץ OK, והאשף יוסיף את המחלקה החדשה לפרויקט וישנה את קובץ Interface Definition Language (IDL), כך שהוא יכלול את ה-CoClass החדשה שנוצרה. קובץ IDL משמש ליצירת ספריית סוגי הרכיב, שהיא חלק COM נדרש. לאחר שהאשף סיים, ייראה הפרויקט שלך דומה לזה שמוצג בתרשים 19.8 (יחד עם המחלקות החדשות שנמצאות בעץ).

## הערה:

למידע נוסף אודות IDL, אני ממליץ על המאמר "Understanding Interface Definition Language: A Developer's Survival Guide" מתוך מהדורת אוגוסט 1998 של Microsoft Systems Journal. תוכל למצוא את המאמר בכתובת האינטרנט http://www.microsoft.com/msj/0898/idl/idltop.htm.

לפרויקט זה, השתמשתי בערך MyClass עבור Short Name בתיבת הדו-שיח ATL Object Wizard Properties. האשף הוסיף שני פריטים לשטח העבודה שלך: CmyClass, מחלקת C++, ו-IMyClass, ממשק שמסופק על ידי האובייקט. ממשק IMyClass הוא המקום בו יוצבו המאפיינים והשיטות עבור CoClass (מייד תקבל מידע נוסף על כך). הממשק נמצא בתוך הגדרת מחלקת C++ בשם CMyClass. הממשק, כפי שהוא נוצר על ידי האשף, מספק שתי שיטות: OnPageStart ו-OnPageEnd. שיטות אלו מופעלות בתחילת ובסוף פענוח ה-Active Server Page שיוצר מופע (instance) מאובייקט חדש זה.



**תרשים 19.8** הפרויקט לאחר שנוסף אובייקט ActiveX Server Component חדש.

**הערה:**

אם נוצר מופע של הרכיב שלך על ידי שימוש בתגית <OBJECT> עם טווח Application, לא תתבצע קריאה לשיטה OnPageStart ו-OnPageEnd משום שטווח הרכיב מגיע מעבר לדף הנוכחי. במקרים מעין אלה, עליך להשתמש בפונקציית GetObjectContext של MTS ולאחזר את האובייקטים הפנימיים בדרך זו. אם אתה סבור שיהיה שימוש כזה ברכיב שלך, כדאי לך להוסיף קוד שמגלה מצבים בהם אין מצביעים תקפים לאובייקטים הפנימיים ולאחזר אותם תוך שימוש בשיטה GetObjectContext.

האשף הוסיף גם כמה משתנים ברמת המחלקה, שיאחסנו את מופעי האובייקטים הפנימיים של ASP (m_piApplication, m_piRequest, m_piResponse, m_piServer, ו-m_piSession), וכן את אובייקט ה-Marshaler מסוג Free (m_pUnkMarshaler). אובייקטי ASP הפנימיים מאוחזרים על ידי קוד שהאשף הציב בשיטה OnPageStart. ה-Marshaler מסוג Free נוצר בפרוצדורה FinalConstruct של המחלקה, והוא משתחרר בפרוצדורה FinalRelease שלה.

למעשה, אתה יכול להדר את הפרויקט בשלב זה ולקבל רכיב ASP, אותו אתה יכול ליצור תוך שימוש ב-()Server.CreateObject. עם זאת, ללא מאפיינים או שיטות, האובייקט לא יוכל לבצע דבר.

# הוספת מאפיינים ושיטות

כמו בכל רכיבי ActiveX, העבודה האמיתית נעשית כשנקבעים מאפיינים או כשמבוצעות שיטות. למרות ש-ATL מספק כמה תיבות דו-שיח פשוטות להוספת מאפיינים ושיטות, הנתונים שאתה מקליד בתיבות אלו מפורט יותר מאשר תיבת הדו-שיח Add Procedure של Visual Basic (ראה בפרק 18). עם זאת, הדרישה להוספת נתונים אלה מאפשרת לך לכוונן בעדינות את ממשק הרכיב שלך בדרכים שלא קיימות ב-Visual Basic.

## הוספת מאפיין

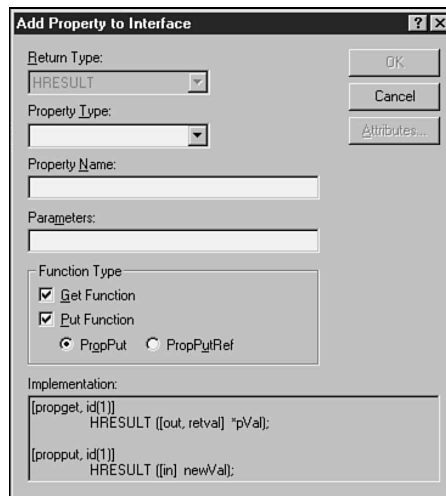
כמו ב-Visual Basic, קל בהרבה להוסיף מאפיינים מאשר שיטות. כשאתה מוסיף מאפיין חדש, אשף ATL Wizard יוסיף כמה פונקציות לממשק המחלקה הגלוי שלך, על פי סוג הגישה למאפיין (קריאה\כתיבה, קריאה בלבד או כתיבה בלבד).

פונקציית הכתיבה תיקרא put_PropertyName ופונקציית הקריאה תיקרא get_PropertyName. בנוסף, אם סוג נתוני המאפיין הוא אובייקט COM נוסף, תהיה פונקציה שלישית, בשם putref_PropertyName, המתבצעת כשקוד הלקוח משתמש ב-Set Object.PropertyName = SomeOtherObjectReference.

למרות שהאשף מוסיף את כותרות הפונקציות לקובץ Include של המחלקה, לסוף קובץ ה-CPP ולקוד המתאים בקובץ ה-IDL, עליך להוסיף את קוד ההטמעה ואולי גם משתנה איבר (Member Variable) שיאחסן את ערך המאפיין בתוך המחלקה שלך.

להלן השלבים להוספת מאפיין מחרוזת למחלקה שלך :

1. תוך שימוש בעץ שבחלונית משטח העבודה, לחץ לחיצה ימנית על פריט הממשק (בתרשים 19.8, זהו IMyClass) המופיע ישר מתחת לפריט השורש ProjectName Classes. מתפריט הקיצור המופיע בחר Add Property.
2. כעת תופיע תיבת הדו-שיח Add Property to Interface, כפי שמוצג בתרשים 19.9. בתיבת דו-שיח זו משתמשים כדי להקליד מידע אודות המאפיין החדש. ראשית, בחר ב-Property Type. תוכל לבחור בסוג מאפיין מהרשימה או להקליד סוג נתוני COM נוסף (כולל ממשקי COM אחרים כגון IFontDisp). עליך לדבוק בערכים שברשימה, פרט לממשקי COM אחרים. בחר ב-BSTR עבור מאפיין מסוג מחרוזת.
3. לאחר מכן, הקלד את שם המאפיין בתיבת הטקסט Property Name. בזמן שאתה מבצע את השינויים בתיבת הדו-שיח, הטקסט במסגרת Implementation משתנה. טקסט זה יתווסף לקובץ IDL עבור המאפיין החדש.




**תרשים 19.9** תיבת הדו-שיח Add Property to Interface

**4.** אם המאפיין משתמש בפרמטרים, הקלד אותם בתיבת הטקסט Parameters. רוב המאפיינים אינם משתמשים בפרמטרים.

**5.** במסגרת Function Type, ציין את הגישה הזמינה למאפיין. נקה את תיבת הסימון Get Function כדי לגרום למאפיין להיות מסוג כתיבה בלבד. נקה את תיבת הסימון Put Function כדי לגרום לו להיות מסוג קריאה בלבד.

**6.** אם סוג הנתונים של המאפיין הוא אובייקט COM נוסף ואתה רוצה לתמוך רק בהצהרת Set, לחץ על לחצן האפשרויות PropPutRef. אם תרצה לתמוך הן בהקצאות ערך מאפיין סטנדרטיות והן מסוג Set, יהיה עליך להוסיף ידנית את הפונקציה put_PropertyName ואת הקוד IDL.

**7.** לחיצה על לחצן Attributes תפעיל את תיבת הדו-שיח Edit Attributes. בתיבת דו-שיח זו, אתה יכול לשנות חלק מתכונות הממשק של המאפיין החדש. ספציפית, עליך לשנות את ערך התכונה helpstring כך שתתאר את המאפיין החדש (ודא שהיא נשארת בתוך גרשיים). תכונה זו מוצגת כטקסט העזרה בדפדפני אובייקטים בכל פעם שהמאפיין נבחר בדפדפן האובייקט.

**8.** לחץ OK בתיבת הדו-שיח Add Property to Interface, והאשף יוסיף את התמיכה הדרושה לפרויקט שלך.

**9.** כדי להגיע למאפיין המוטמע, עליך להרחיב את שם המחלקה בחלונית משטח העבודה. כשתעשה זאת, תראה את שם הממשק כצומת צאצא (Child Node). הרחב את הצומת, ותראה את הערכים של פונקציות get ו-put של המאפיין החדש. לחץ לחיצה כפולה על הערך, וחלון הקוד יטען את הפונקציה. מה שתראה יהיה דומה למה שמוצג בתדפיס 19.1.

**תדפיס 19.1** הקוד שנוסף למאפיין מחרוזת חדש

```
STDMETHODIMP CMyClass::get_Name(BSTR *pVal)
{
AFX_MANAGE_STATE(AfxGetStaticModuleState())

    // TODO: Add your implementation code here

return S_OK;
}

STDMETHODIMP CMyClass::put_Name(BSTR newVal)
{
AFX_MANAGE_STATE(AfxGetStaticModuleState())

    // TODO: Add your implementation code here

return S_OK;
}
```

**10.** כדי להוסיף משתנה איבר שיאחסן את ערך המאפיין, לחץ לחיצה ימנית על שם המחלקה ומתפריט הקיצור בחר Add Member Variable. כעת תופיע תיבת הדו-שיח Add Member Variable. הקלד את סוג הנתונים (CComBSTR במקרה הזה) ואת שם המשתנה הרצוי (כמובן, על פי הרגליך). תוכל לשלוט בגישה למשתנה האיבר תוך שימוש בלחצני האפשרויות במסגרת Access. לחץ OK, והמשתנה יתווסף למחלקה שלך.

**11.** כעת אתה יכול לשנות את קוד המאפיינים המוטמע, כך שישתמש במשתנה האיבר החדש (ראה תדפיס 19.2).

**תדפיס 19.2** קוד תמיכה במאפיין המחרוזת

```
STDMETHODIMP CMyClass::get_Name(BSTR *pVal)
{
AFX_MANAGE_STATE(AfxGetStaticModuleState())

if (pVal = NULL) return E_POINTER;
*pVal = m_bstrName.Copy();

return S_OK;
}

STDMETHODIMP CMyClass::put_Name(BSTR newVal)
{
AFX_MANAGE_STATE(AfxGetStaticModuleState())
m_bstrName = newVal;
return S_OK;
}
```

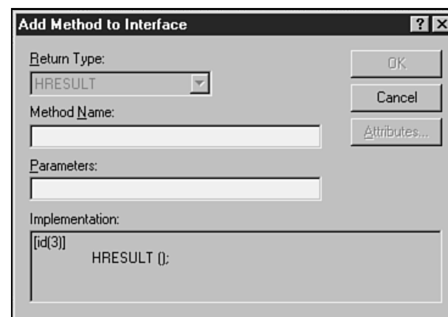
**הערה:**

קובץ IDL נוצר למטרה גלובלית בתפעול רכיבים על ידי מנועים כגון Visual Basic, VBScript ו-ASP. קובץ זה בעצם מספק מידע על כל השיטות והמאפיינים של הרכיב בצורה כללית עבור כל שפה שנתמכת ב-Windows.

# הוספת שיטה

תהליך הוספת שיטה לרכיב דומה למדי לתהליך הוספת מאפיין. ההבדל המשמעותי ביותר הוא שאתה חייב להקליד ידנית את חלקי ה-IDL כשאתה מציין את פרמטרי השיטה. להלן השלבים:

1. תוך שימוש בעץ שבחלונית משטח העבודה, לחץ לחיצה ימנית על ערך הממשק (בתרשים 19.8 זהו IMyClass) המופיע ישר מתחת לערך השורש ProjectName Classes. כעת יופיע תפריט קטן של קיצורי דרך. לחץ על הפריט Add Method.
2. כעת תופיע תיבת הדו-שיח Add Method to Interface (ראה תרשים 19.10). הקלד את שם השיטה בתיבת Method Name.




**תרשים 19.10** תיבת הדו-שיח Add Method to Interface

3. לאחר שהקלדת את השם, לחצן Attributes הופך לזמין. כמו במאפיינים, לחצן זה מציג תיבת דו-שיח להקלדה ולשינוי תכונות הממשק של השיטה. לחץ על הלחצן כדי לשנות את תכונת helpstring למשהו שמזכיר יותר את מטרתה.
4. בתיבת הטקסט Parameters, הקלד את כל הפרמטרים המועברים לשיטה, וכן פרמטר לטיפול בערך החוזר שלה (במידה ונדרש פרמטר כזה). דבר זה נעשה בפורמט IDL. לדוגמה, [in] long InitialValue, [in,out] long * Multiplier, [out, retval] long* returnValue.

הפרמטר הראשון, [in] מציין שהפרמטר עובר על ידי ערך, כלומר, אם נעשים שינויים לפרמטר על ידי קוד השיטה, הם לא ישתקפו במשתנה אותו העביר החלק הקורא.

הפרמטר השני מצוין בעזרת [in, out] וכ**מצביע** (Pointer), כלומר, כתובת המשתנה בצד הקורא) למשתנה. המשתנה יעביר את השינויים שעשויים להתקבל בחזרה לקורא. למרות ששפות תסריט ו-Visual Basic ימירו אוטומטית משתנה למצביע שלו, כל קוד Visual C++ שמשתמש בשיטה זו יהיה חייב לציין במפורש שיש להשתמש בכתובת.

הפרמטר האחרון משתמש ב-[in, out] ובמצביע המספק את ערך ההחזרה. קוד התסריט אינו מספק ערך זה כפרמטר ישיר, אלא משתמש בשם השיטה בהקצאה, כגון x = object.method(1, lMultiplier). קוד C++ שמטמיע שיטה זו לא ישתמש **בערך ההחזרה** כדי לקבוע את ערך פרמטר זה, אלא יקבע אותו במפורש, כגון *returnValue = InitialValue * Multiplier.

5. לחץ OK והאשף יוסיף את הקוד המתאים לקובץ IDL, לקובץ כותרות המחלקות (Class Header) ולקובץ CPP של המחלקה. תדפיס 19.3 מציג את מה שיוצב בקובץ CPP.

**תדפיס 19.3** הקוד שנוסף עבור שיטה חדשה

```
STDMETHODIMP CMyClass::GetValue(long InitialValue
                    long *Multiplier, long *returnValue)
{
   AFX_MANAGE_STATE(AfxGetStaticModuleState())

   //  TODO: Add your implementation code here

   return S_OK;
}
```

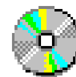
6. כל שנשאר לעשות הוא להוסיף את הקוד לשיטה שלך.

## הסרת מאפיין או שיטה

לרוע המזל, ל-ATL אין אשף להסרת מאפיין או שיטה. כדי לעשות זאת, אתה חייב להסיר את ההתייחסויות למאפיין או השיטה משלושה מקומות. ראשית, הסר את הקוד המוטמע מקובץ CPP של המחלקה. שנית, הסר את הגדרות הפונקציה מקובץ הכותרת של המחלקה. לבסוף, הסר את ההתייחסות למאפיין או השיטה מתוך קובץ IDL.

זכור שניתן להטמיע מאפיין עם יותר מפונקציה אחת, כתלות בהרשאות הגישה אליו. אם למאפיין יש הרשאת כתיבה, תהיה לו פונקציית put_PropertyName ו\או putref_PropertyName וכן ערכי IDL עבור כל אחת. אם למאפיין יש הרשאת קריאה, תהיה לו פונקציית get_PropertyName וערך IDL.

# דוגמה פשוטה



כעת תיצור דוגמה פשוטה שיש לה מאפיין אחד ושיטה אחת. הרכיב ישמש להמרת ערך למחרוזת שמעוצבת תוך שימוש בהגדרות המטבע של השרת (כפי שהוגדרו על ידי היישומון Regional Settings שבלוח הבקרה). הגירסה הסופית של כל הקבצים שקשורים לרכיב זה נמצאת בתקליטור המצורף בתיקיה Chapter19\formatter.

לרכיב זה מאפיין אחד, Value, שמוגדר כ-double. היית יכול להשתמש בסוג נתונים CURRENCY, אך עליך לאפשר תחום רחב יותר של מספרים, למקרה שטווח הרכיב כולל סוגי עיצוב אחרים. לרכיב שיטה אחת, FormatAsCurrenty. לשיטה זו אין פרמטרים נכנסים אך היא מחזירה מחרוזת (שמוגדרת ב-COM תוך שימוש בסוג נתונים BSTR). המחרוזת תעוצב כך שניתן יהיה להכניס אותה לקוד HTML על ידי שימוש בשיטה Server.HTMLEncode(). ה-ProgID בו משתמשים עבור רכיב זה יהיה Unleashed.Formatter.

להלן הדרך ליצירת הרכיב:

1. התחל פרויקט ATL COM AppWizard חדש. קרא לפרויקט בשם Unleashed. אל תוסיף תמיכת MFC אלא אם כן אתה רוצה להכניס תוספות תוך שימוש ב-MFC.
2. הוסף מחלקת ActiveX Server Component חדש על ידי שימוש באפשרות New ATL Object בתפריט Insert.
3. בכרטיסיה Names, הקלד את הערך Formatter בתיבת הטקסט Short Name.
4. בכרטיסיה Attributes, ציין את האפשרות Apartment Threaded (אלא אם כן אתה מתכנן לאחסן מופע (instance) של הרכיב ב-Session Variable. במצב כזה השתמש במודל Both וסמן את תיבת הסימון Free-Threaded Marshaler). בנוסף, סמן את תיבת Support ISupportErrorInfo.
5. בכרטיסיה ASP, בטל את סימון ערכי האובייקטים הפנימיים, פרט לערך Server. לחץ OK כדי להוסיף את המחלקה.
6. הרחב את העץ שבחלונית משטח העבודה ולחץ לחיצה ימנית על הערך Iformatter. מתפריט הקיצור בחר באפשרות Add Property. השתמש בסוג נתונים double וקרא למאפיין בשם Value. השאר את כל הערכים האחרים בברירת המחדל שלהם, אלא אם אתה רוצה לשנות את תכונת helpstring (אותה תמצא על ידי לחיצה על Attributes). לחץ OK כדי להוסיף את הגדרת המאפיין והפונקציות התומכות בו למחלקה.
7. לחץ לחיצה ימנית על הערך Cformatter ומתפריט הקיצור בחר Add Member Variable. בתיבת הדו-שיח Add Member Variable, הקלד את הערך double כסוג הנתונים ואת הערך m_dblvalue כשם המשתנה. לחץ OK.
8. הרחב את הערך Cformatter בחלונית משטח העבודה, הרחב את ערך Iformatter שלו ולחץ לחיצה כפולה על הערך get_Value. שנה את קוד הפונקציה הזו ופונקציית put_Value כך שיתאימו לתדפיס 19.4.

**תדפיס 19.4** קוד מאפיין Value

```
STDMETHODIMP CFormatter::get_Value(double *pVal)
{
    if (pVal == NULL) return E_POINTER;

    *pVal = m_dblValue;
```

```
    return S_OK;
}

STDMETHODIMP CFormatter::put_Value(double newVal)
{
    m_dblValue = newVal;

    return S_OK;
}
```

**9.** לאחר מכן, הוסף את השיטה FormatAsCurrency. לחץ לחיצה ימנית על הערך Iformatter ומתפריט הקיצור בחר Add Method. הקלד את הערך FormatAsCurrency בתיבת הטקסט Method Name. הפרמטר היחיד של השיטה הוא ערך ההחזרה שלה, לכן בתיבת הטקסט הריקה Parameters הקלד את הערך [out, retval] BSTR* returnVal. לחץ OK, והשיטה תתווסף.

**10.** לחץ לחיצה כפולה על ערך השיטה המופיע בעץ שבשטח העבודה והקלד תדפיס 19.5.

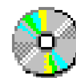
**תדפיס 19.5** קוד השיטה FormatAsCurrency

```
STDMETHODIMP CFormatter::FormatAsCurrency(BSTR *returnVal)
{
    USES_CONVERSION;

    char pszTemp[MAX_PATH];
    char pszOutput[MAX_PATH] = {0};

    sprintf(pszTemp, "%f", (float)m_dblValue);

    int iRet = GetCurrencyFormat(LOCALE_SYSTEM_DEFAULT,
LOCALE_NOUSEROVERRIDE,
                        pszTemp, NULL, pszOutput, MAX_PATH);

    if (iRet) {

        if (m_bOnStartPageCalled) {

            CComBSTR bstrTemp(pszOutput);
            m_piServer->HTMLEncode(bstrTemp, returnVal);

        } else

            *returnVal = ::SysAllocString(A2W(pszOutput));

    } else
        *returnVal = NULL;

    return S_OK;
}
```

אם אינך מתמצא ב-C++ או בסוג הנתונים BSTR, להלן תיאור בסיסי של דרך פעולת השיטה. המאקרו USES_CONVERSION קובע קוד מסוים הנדרש לקריאה למאקרו A2W. פונקציית ה-API GetCurrencyFormat משמשת לעיצוב הערך. הדרישה לפונקציה זו היא שהערך יוצב במערך תווים. ניתן לעשות זאת תוך שימוש בפונקציות pszTemp ו-sprintf. הפונקציה מחזירה את המחרוזת המעוצבת לתוך המאגר pszOutput. הפרמטרים LOCALE_SYSTEM_DEFAULT ו-LOCALE_NOUSEROVERRIDE מציינים את הדרך בה ייקרא העיצוב מתוך ההגדרות האזוריות. הפונקציה מחזירה את מספר התווים שהוכנסו למאגר (Buffer).

אם ישנם תווים מוחזרים, הקוד בודק אם ערך משתנה m_bOnStartPageCalled הוא true. הערך יהיה true אם השיטה OnStartPage הופעלה (במילים אחרות, אתה בסביבת ASP, ולא במופע של הרכיב שנוצר על ידי Visual Basic), והמצביע למופע (instance) האובייקט הפנימי Server תקף. אם זהו המצב, אתה יכול לסמוך על כך ש-m_piServer מצביע על מופע של האובייקט ASP הפנימי Server. קרא לשיטה שלו HTMLEncode כדי לוודא שניתן יהיה להציג את המחרוזת החוזרת ממנו בדף HTML.

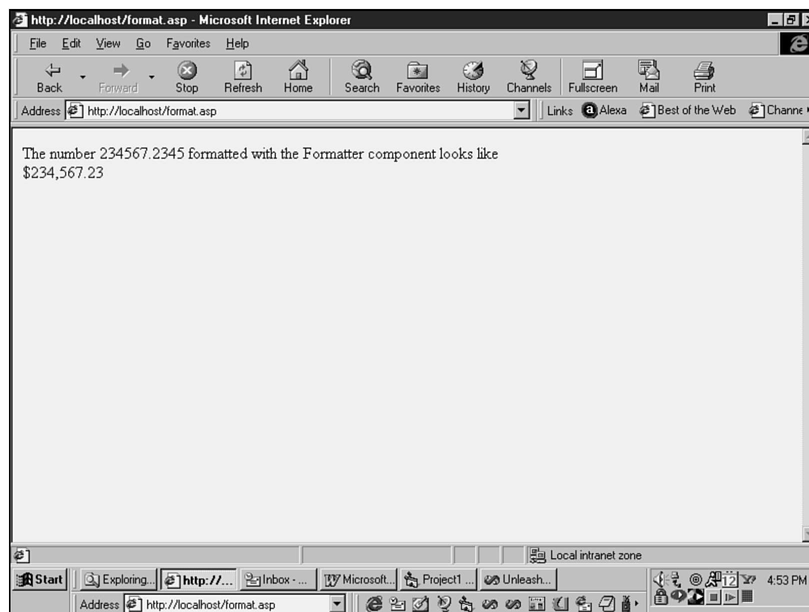
אם ערך משתנה m_bOnStartPageCalled אינו true, אינך יכול להשתמש באובייקט m_piServer. לחילופין, ערך המצביע returnVal מצביע על תוצאת SysAllocString. SysAllocString לוקחת wchar וממירה אותו ל-BSTR וכן מקצה את השטח הדרוש כדי לאחסן את המחרוזת. המאקרו A2W ממיר מערך char למערך wchar, הדרוש עבור SysAllocString.



הדר את הפרויקט, והרכיב מוכן לשימוש. בפרויקט ATL COM AppWizard, המהדר דואג לרשום את רכיבי ה-ActiveX במחשב שלך. תדפיס 19.6 מציג Active Server Page שמבצע תרגול על הרכיב החדש. תוכל למצוא את הקובץ format.asp בתיקיה Chapter19\formatter בתקליטור המצורף. תרשים 19.11 מציג את פלט הדף.

**תדפיס 19.6** קוד ה-ASP הדרוש להפעלת הרכיב החדש

```
<%@ Language=VBScript %>

<HTML><BODY>
The number 234567.2345 formatted with
the Formatter component looks like<br>
<%
   set xFormatter = Server.CreateObject("Unleashed.Formatter")
   xFormatter.Value = 234567.2345
   Response.Write xFormatter.FormatAsCurrency
%>
</BODY></HTML>
```




**תרשים 19.11** הפלט המתקבל מדף ASP המשתמש ברכיב Formatter.

# סיכום

פרק זה דן בעקרונות יצירת רכיבי דף ASP תוך שימוש ב-++Visual C. כשאתה יוצר רכיבים בעזרת ++Visual C, אתה מקבל את הגמישות הרבה ביותר, ובשל כך גם את הרכיבים המהירים ביותר. עם זאת, למתכנת יש קצת יותר עבודה ביצירת הרכיב. מודלי המטלות השונים תוארו, ולמדת שהמודלים המתאימים עבור רכיבי ASP הם apartment או both.

לאחר מכן, ראית כמה קל להשתמש ב-ATL ובאשפים שלו כדי ליצור רכיב חדש, להוסיף את המאפיינים והשיטות שלו, ולהקליד את הקוד הדרוש לביצוע העבודה הממשית. לבסוף, יצרת רכיב דוגמה קטן אך שימושי שממיר מספר לפורמט מטבע (Currency format) בהתבסס על ההגדרות האזוריות של השרת.

# חלק 5

# יצירת יישומים מבוזרים

**בחלק זה:**

- **הבנת Windows DNA**
- **שימוש ב-ASP יחד עם Microsoft Transaction Server**
- **שימוש ב-ASP יחד עם Microsoft Message Queue Server**

## פרק 20

# הבנת Windows DNA

**בפרק זה:**

- **חזון Windows DNA**
- **השירותים של Windows DNA**
- **כיצד משתלב ASP בחזון זה**

עד לנקודה זו בספר, כל הדברים שנגעו ל-ASP היו כולם מכניים בלבד, כלומר, היו הוראות בסגנון "אם תקבע את פרמטר X ותקרא לשיטה Y, משהו יקרה". בנוסף לכך, היו כמה הדגמות קטנות שהראו כיצד ניתן להשתמש בדברים המכניים הללו כדי לבצע פעולה.

כעת נשנה את הגישה ונראה כיצד הכל משתלב יחד לצורך בניית יישום שמטרתו לפתור בעיות. פרק זה מסביר מהו ה"חזון" הזה. במונחי Microsoft, המשמעות היא Windows DNA. ההסבר לגבי Windows DNA לא יהיה מכני או טכני, אלא ינסה לאפשר לך לראות את התמונה הגדולה. הסעיף הבא מפרט כמה עניינים טכניים שנוגעים לשירותים השונים שבחזון. לבסוף, תקבל הסבר על תפקידו של ASP בחזון Windows DNA.

**הערה:**

DNA הוא קיצור של **D**istributed inter**N**etwork **A**pplication (בין השאר).

במהלך פרק זה נעשה שימוש במילה **לקוח** (client) כדי לתאר בן-אנוש והן כדי לתאר מחשב.

# חזון Windows DNA

כדי להבין כל תפישה שהיא, עליך להבין את ההקשר שלה, וכיצד הגיעו אליה. לפיכך, כדי להבין את חזון Windows DNA, עליך להבין כיצד התפתחה פלטפורמת Windows עצמה.

- Windows 3.x - הביאה משהו חשוב יותר לזירה. היא הביאה ממשק משתמש נפוץ ומודל מנהלי התקנים (Drivers). לפני עידן Windows, היה צורך לכתוב מנהלי התקנים (Drivers) נפרדים לכל יישום DOS, פעולה מייגעת. ליישומים בעלי מנהלי התקנים (Drivers) מותאמים, היו דרישות זיכרון ספציפיות ובעיות משלהם. Windows שינתה את פני הדברים ואיפשרה ללקוח לדרוש מנהלי התקן (Drivers) של Windows עצמה. ממשק המשתמש ויכולת ביצוע המשימות היו טובים אף הם, משום שהם היו מסוגלים לבצע בעקביות משימות מסוימות.
- Windows NT - ל-Windows 3.x היתה בעיה של חוסר יציבות. Windows NT היתה מערכת הפעלה חדשה שנבנתה עבור ארגונים. בדרך מסוימת, היא היתה גירסה יציבה של Windows 3.x. NT כללה כמה דברים, כגון שילוב תכונות רשת ורכיבי שרת, שנקראו **שירותים** (Services). אך רעיון שירותי השרת לא היה רחב מספיק כדי לאפשר כתיבת יישומי צד-שרת, משום שהוא לא סיפק אפשרות תנועות (Transactions), הודעות (Messaging) וכדומה.
- Windows 95/98 - משפחת מערכות Windows המשיכה לצמוח גם כשהוצגו החלופות ל-Windows 3.x. Windows 95 ו- Windows 98 הן מערכות הפעלה שפועלות בסביבת 32 סיביות. עם זאת, הן שונות מ-Windows NT בשוק היעד שלהן. Windows 95/98

מיועדות לצרכנים האישיים. במערכות אלו הדרישה היא להתקנים שיודעים להגדיר את עצמם ומקלים על ניהול מחשב הלקוח. משימה זו מרתיעה משום שישנם מיליוני מחשבים שפועלים בסביבת מערכות Windows. Windows 95/98 מתמקדות בקלות על הניווט באינטרנט וביצוע המשימות הספציפיות.

- Windows NT - Windows NT Option Pack מספקת את הבסיס למערכת הפעלה יציבה הניתנת להרחבה. המערכת לא מספקת את השירותים הנוספים שנדרשים לארגון. Windows NT Option Pack מוסיפה את השירותים הללו. חבילת ההרחבה מכילה שירותים כגון IIS (Internet Information Server), MSMQ (Microsoft Message Queue), MTS (Microsoft Transaction Server) ועוד. כשיצאה לאור Windows Option Pack החלו מאמצים לפיתוח שירותים ארגוניים ב-Microsoft. דבר זה ציין את השוני בין מפתחי התוכנה הפרטית למפתחי התוכנה הארגונית שנדרשה עבור חברות.
- Windows 2000 - יציאתה של Windows 2000 לאור בישרה על שירותים נוספים, ועל COM עקבי ואלגנטי יותר שנקרא בשם +COM. Windows 2000 היא מערכת הפעלה שמסוגלת לטפל במשימות גדולות בארגונים גדולים.

# מהו Windows DNA?

Windows DNA הוא מהות מערכת ההפעלה שחושפת שירותים ותכונות מנגנונים אשר מאפשרים לבנות יישומים תוך שימוש בתהליך תכנון לוגי.

הגדרה זו מטושטשת למדי, אך היא ספציפית. לפי ההגדרה, עליך להשתמש בתהליך תכנון לוגי לבניית יישומים. פעמים רבות מדי בתהליך הפיתוח אנו נתקלים במחסום, מפני שאין בידינו עדיין את הטכנולוגיה הדרושה. כדי לפתור את הבעיה, אנו יוצרים לעצמנו פתרונות אישיים, או כאלה שנלקחים מדברים אחרים שכבר פותחו. פתרונות אישיים אלה גורמים לבעיות, משום שהם החוליה החלשה בשרשרת. כשהיישום הופך למורכב יותר, השמירה על החוליות החלשות האלו קשה ומייגעת יותר.

לדוגמה, פרויקט שהייתי מעורב בו כלל יישום חדש לגמרי בממשק המשתמש של פלטפורמת Windows (Windows GUI). היישום הישן נכתב ב-DOS ובאסמבלר. היישום היה צריך לבצע תקשורת לשרת והוא עשה זאת תוך שימוש בספריית Socket מותאמת (Socket הינו מתווך דרכו מגיע מידע ליישום ודרכו מועבר מידע לשרת). היישום החדש היה חייב לעבוד עם DOS, אסמבלר, וממשקי Socket. ביישום זה בלבד, היו שלושה מנגנוני תקשורת. Windows DNA פתר זאת על ידי הגדרת התקשורת כשירות (Service) בשם COM, שהיתה לו שכבה אחת והיה קל להתקין אותו (Plug And Play). אבן היסוד של Windows DNA היא COM ובניית רכיבים.

# הבעיות העומדות מול Windows DNA

הבעיה אותה Windows DNA פותר היא היישום. בכל חברה יש תהליכים עסקיים (Business processes). דוגמאות לכך כוללות יצירת טפסי הזמנה, הפקת חשבוניות, והצגת נתוני רווח. יישום זה צריך לפעול בתרחיש מרובה משתמשים ברשת, ולרוב גם באינטרנט.

לפיכך, ליישום יהיו התכונות הבאות:

- הוא יתקשר עם המשתמש תוך שימוש במקלדת, מסך או עכבר.
- ייתכן והוא ישתמש בהתקני קלט אחרים כמו מסך מגע, עט אלקטרוני והפעלה קולית.
- הוא יתחבר למחשבים אחרים תוך שימוש בתפישות הרשת.
- כשכמה מחשבים משתפים ביניהם נתונים, הוא חייב לפעול בצורה אמינה ועקבית.
- היישום חייב לדעת לנהל את עצמו ואת הסביבה שלו בצורה קלה.

זהו שינוי גדול ממה שהיה בימי ה-API של Windows. באותה תקופה, הדגש היה על מראה טוב של ממשק המשתמש. השינוי בהתמקדות נובע מכך שהמחשב הוא אבן יסוד בחברה שלנו. איננו יכולים לחיות בלעדיו. לאנשים יש את היכולת ללמוד מחשבים ולפעול איתם, אך הם אינם מעוניינים להתעמק בפרטי תוכניות ההתקנה והתסריטים שלהן, פרמטרים של שורת פקודה, ונפילות יישומים. אותם אנשים רוצים שהכל יפעל כשורה.

## הפתרון: Windows DNA

פלטפורמת Windows DNA בנויה על פי מודל יישום מפוצל מרובה דרגות (Multitiered) (אני אפרט יותר בנוגע לפלטפורמה בסעיף הבא). היא מספקת ערכה של שירותים ותכונות שמהוות חלק מתשתית מערכת ההפעלה. שירותים אלה משלבים דברים כגון מחשבים אישיים, מחשבים מרכזיים, אינטרנט, לקוח/שרת ועוד.

כאשר כל הטכנולוגיות הללו מעורבות, מערכת ההפעלה עלולה להתנפח מדי, אך בפועל המצב אינו כזה משום שאתה יכול לבחור רק בשירותים הדרושים לך. דבר זה נעשה על ידי בחירה בדרגת Windows 2000 המתאימה והתקנת השירותים הנחוצים. העקרונות המנחים של Windows DNA הם:

- מוכנות לאינטרנט - טכנולוגיות אינטרנט, כמו TCP/IP, HTTP ו-FTP, משולבות בתוך מערכת ההפעלה. בשל כך, היישומים יכולים לסמוך על זמינות אינטרנט קיימת.
- פיתוח יישומים מותאמים אינו פשוט. יישום מותאם מכוונן במיוחד עבור הלקוח. דוגמה לכך הוא יישום בנקאי. כדי שהבנקים יהיו יעילים יותר, הם דורשים יישומים שמותאמים במיוחד לסביבתם הם. פתרונות אלה מתאימים רק לתעשיית הבנקאות, ולרוב הם מבוצעים רק אצל לקוח יחיד. אם אתה יכול להתמקד בכתיבת יישום תוכל להגיע לשוק מהר יותר.

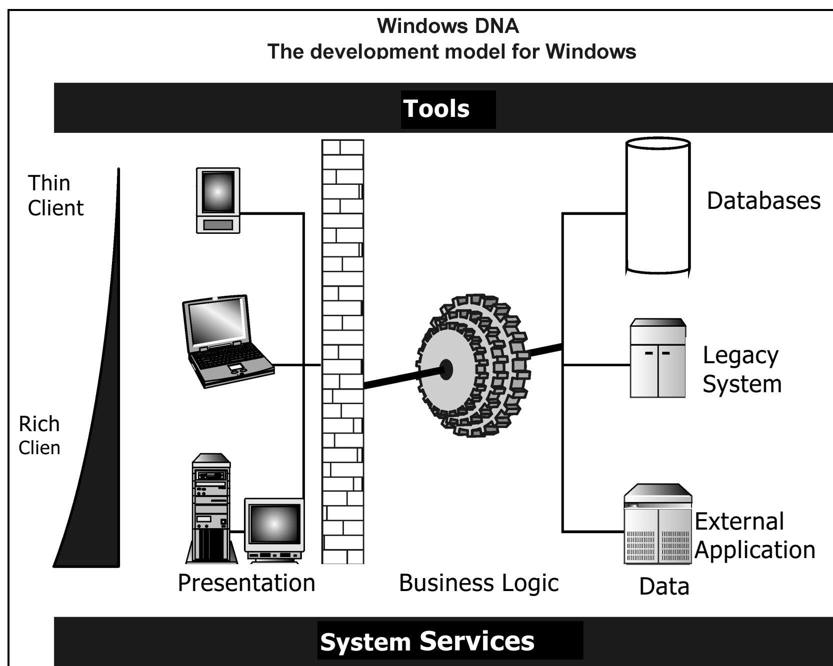
- יכולת עבודה הדדית - נושא ריבוי הפלטפורמות מורכב. אין לכך פתרון קל. דרך נוספת להשיג כמעט את אותו הדבר היא היכולת לפעול בצורה הדדית. יכולת זו משתמשת בפרוטוקולים וסטנדרטים פתוחים כדי לוודא שהפתרון יפעל עם מוצרים של מספר יצרנים.
- פחות מורכבות - כשיש כמה התקנות, העניינים מסובכים יותר. כמה מהדברים הללו הם חלק ממערכת ההפעלה. דוגמה לכך היא TCP/IP. העתיד טמון ברשת TCP/IP, ולכן הגיוני לשלב אותה במערכת ההפעלה. דבר זה הופך את ההתקנה כולה ואת תא נושאי התמיכה לפחות מכבידים משום שיש פחות משתנים שעליך לקחת בחשבון. שוב, דבר זה מאפשר לך להתמקד בפתרון הבעיה העסקית (Business problem).
- עצמאות כלי השפה - בסיס פלטפורמת Windows היא לאפשר למפתחים נוספים לבחור בסביבות הפיתוח שלהם. הם יוכלו לבחור בשפה שמתאימה ביותר לפתרון בעיית התהליך העסקי (Business process problem).
- הנמכת העלות הכוללת של הבעלות - השלב האחרון הוא להקל על פריסה, ניהול ומחזור היישום על פני הזמן. ניתן להשיג זאת על ידי פישוט תהליכי ההתקנה ובקרת הגרסאות.

לאחר שקראת את העקרונות המנחים הללו, אתה עשוי לקבל את הרושם ש-Windows DNA מיועד רק לארגונים גדולים שיש בהם כמה מערכות הפעלה ואלפי לקוחות. למעשה, DNA טוב הן לחברות קטנות והן לגדולות. DNA חייב להתאים לשני הסוגים, משום שלעיתים חברה קטנה יוצרת קשר עם חברה גדולה, ואז הן חייבות לתקשר ביניהן תוך שימוש באותה שפה בינארית.

על ידי שימוש בטכנולוגיות אינטרנט, ניתן להחליף מידע עם מיליוני אנשים אחרים. בתרחיש זה, השירותים חייבים להיות זמינים 24 שעות ביממה. בעניין מעורב יותר מאשר אזור זמן יחיד, והגישה ליישומים תתבצע בכל שעות היממה.

# השירותים של Windows DNA

תרשים 20.1 מציג את ארכיטקטורת Windows DNA, בה משתמשים כדי לבנות יישומים מדורגים. ישנן שלוש דרגות: תצוגה, לוגיקה עסקית (Business logic) ונתונים. זוהי אינה ארכיטקטורה פיסית. היא מייצגת תפישה מופשטת. ייתכן מצב בו כמה דרגות פיסיות יתקיימו בתוך דרגה מופשטת אחת.

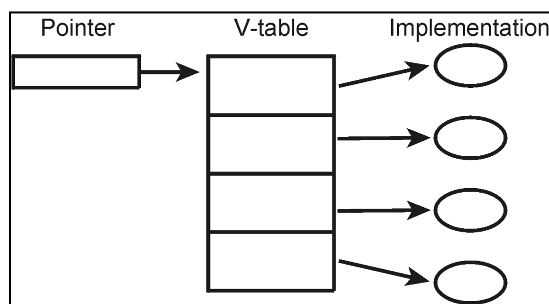



**תרשים 20.1** ארכיטקטורת Windows DNA

# COM בכל מקום

נקודת המפתח ליצירת ארכיטקטורה זו היא לאפשר לכל שכבה לתקשר עם חברתה. דבר זה נעשה על ידי שימוש ב-COM (Component Object Model). זהו מודל אובייקט בינארי שמגדיר שיטות מוגדרות מראש להחלפת נתונים. כשמודל זה חודר לנבכי היישום, השימוש בשירותים השונים והתקשורת בין אובייקטי ה-COM פשוט מאוד.

למרות שציינתי ש-COM הוא בינארי, הוא פועל על מיגוון פלטפורמות. התבונן בתרשים 20.2. מצביע (Pointer), שהוא חלק מהזיכרון שמכיל כתובת בה מאוחסנים נתונים אחרים, מתייחס ל-vtable. **vtable** היא מקטע זיכרון שמכיל סדרה של מצביעים אחרים. עם זאת, הפעם המצביעים מכוונים לפונקציות, אשר מבצעות פעולות רצויות.




**תרשים 20.2** ממשק COM וטכניקת הטמעה

דרך זו של עיצוב אובייקטי COM נקראת הפרדת הממשק מהביצוע. בהגדרת ממשק, ניתן להגדיר מה הוא אמור לעשות. היישום הוא הבנת כוונה זו. בפרק הקודם, בו בנית אובייקטי COM שהתבצעו אליהם קריאות על ידי ASP, עשית זאת במרומז.

COM פועל על פלטפורמות רבות משום שהממשק הוא הגדרה סטנדרטית בכל הפלטפורמות. מחרוזת COM שנקראת BSTR זהה למחרוזת BSTR במעבדי Intel, Sparc ו-Alpha. הדבר שאינו פועל על כל הפלטפורמות בצורה זהה הוא היישום הבינארי משום שהוא מהודר ספציפית עבור כל פלטפורמה בנפרד.

כדי להשיג ממשק ויישום מרובי פלטפורמות ניתן להשתמש בטכנולוגיה כגון Visual J++, מפני שהקוד הפנימי של Java פועל בפלטפורמות רבות. בשימוש ב-Java Virtual Machine (VM) של Microsoft, התואמת ל-COM, עליך לכתוב את הקוד ולהדר אותו פעם אחת בלבד.

אם כן, מדוע להשתמש ב-COM? פרט לסיבות הטכניות, ה-COM הנוכחי הוא ללא ספק אסטרטגיית הרכיב הגדולה ביותר. מסיבה זו, DNA מבוסס על COM. לא ניתן לשבור את התאימות. בעזרת COM, ניתן לבנות רכיבי תוכנה אותם אפשר להפעיל ולפרוס בכל אחת מהדרגות. מודל הפעולה (Runtime) של COM תומך ב- Packaging, Partitioning ונושאים אחרים שקשורים בריבוי דרגות.

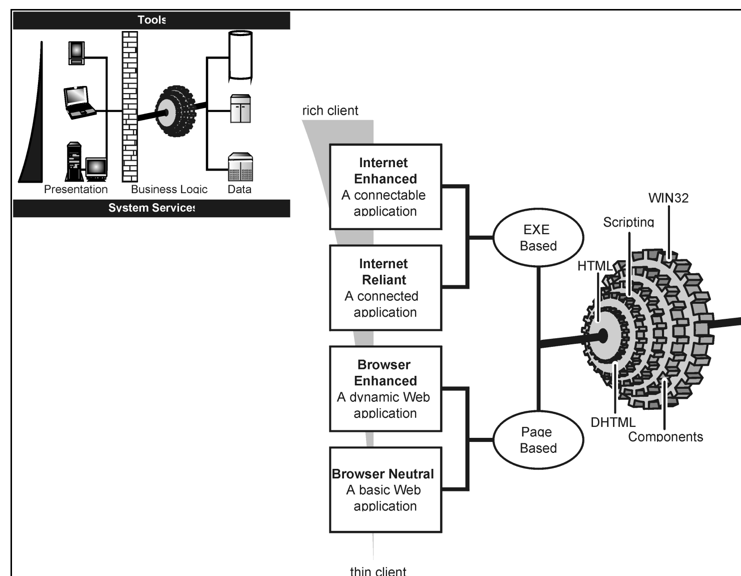
## Presentation Tier

דרגת התצוגה (Presentation Tier) היא הדרגה בה היישום יוצר קשר עם בן אנוש. דרגה זו נחוצה משום שלבני האדם אין את היכולת להבין את משמעות הסיביות והבתים שבדיסק הקשיח בצורה יעילה. בדרגה זו, הנתונים מוצגים בדרך בה נוכל להבין אותם. נכון להיום, דרך זו מבוססת בעיקר על המקלדת, מסך ועכבר. כל זה ישתנה בעתיד כדי לספק התקנים שיסייעו לאנשים עם מגבלות - התקני קול ותנועות.

בגרסת DNA הנוכחית של Windows, ההתמקדות בדרגת התצוגה היא לקדם גישה מכל מקום, בכל צורה ובכל זמן שהוא. דבר זה אפשרי על ידי שילוב האינטרנט בדמות דפדפן, חיוג, ותכונות רשת. דרגה זו מהווה אתגר בשל סוגי הלקוחות הרבים. בארכיטקטורת Windows DNA, הלקוח לא מתבסס רק על Windows, אלא יכול גם להתבסס על Unix, מקינטוש ומערכות הפעלה אחרות.

כדי לאפשר תמיכה בפלטפורמות השונות, דרגת התצוגה צריכה לעבור מ-rich client ל-thin client. אל תחשוב ש-thin client משמעו פחות בתים להורדה משום ש-thin client עשוי לכלול תמונה בגודל של כמה מאות מגה בית. הפרמטר המשתנה הוא כמות הפונקציונליות שניתן להשתמש בה בצד הלקוח.

מה אנו רואים בתרשים 20.3? זהו הדגם המופשט של יצירת דרגת התצוגה. בצד הימני, נראות הטכנולוגיות השונות המיוצגות על ידי סדרת שיניים בגלגל. על ידי שימוש בטכנולוגיה, אנו יכולים לבנות שני סוגים של לקוחות (לקוח מבוסס EXE ולקוח מבוסס דף). ניתן לחלק הלאה סוגים אלה של לקוחות לארבעה סוגי לקוחות, המדורגים החל מ-thin client ועד rich client.




**תרשים 20.3** פרטי דרגת התצוגה

# פתרונות מסוג Page-based

thin client הוא סוג של לקוח מבוסס דף (Page-based Client), כלומר, הלקוח מחובר לרשת והוא מוריד ממנה תוכן על פי צרכיו. הוא אינו מוריד את היישום במלואו, אלא רק דפים מתוכו.

ספציפית, thin client הוא כזה משום שהוא מציג נתונים סטטיים שנקבעו לדרגת התצוגה. לאחר שהדף מתקבל, המשתמש יכול למלא כמה תיבות טקסט, לסמן כמה תיבות סימון, ולהעלות אותו בחזרה לעיבוד. סוג פונקציונליות זה דומה לתהליך עיבוד טפסי אצווה (Batch form Processing) שהיה מקובל בימי מחשבי ה-Mainframe. ההבדל היחידי הוא שכיום המסוף יכול להיות בכל מקום באינטרנט. לרוב, סוג לקוח זה הוא דפדפן אינטרנט. גישת דפדפן אינטרנט סטטי תבטיח שהיישום שלך יקבל את התמיכה הרחבה ביותר. סוג יישום זה נקרא יישום **דפדפן** (Browser-neutral). בסוג פתרון זה, סביבת ההפעלה נקבעת על ידי הלקוח. אחד מיישומי הדפדפן הנפוצים הוא האינטרנט.

אם יש צורך בפונקציונליות נוספת, עליך להשתמש ביישום אינטרנט דינמי. במצב זה, אתה עושה שימוש בממשק משתמש לאינטרנט מתקדם יותר. דבר זה מאפשר יצירת יישומים אינטראקטיביים יותר. עם זאת, מחיר האינטראקטיביות המשופרת הוא בכך

שפחות דפדפני אינטרנט מסוגלים לפענח את התוכן. לדוגמה, ניתן לפתח תוכן שיתפקד רק ב-Internet Explorer או ב-Netscape Navigator. יתרון הפונקציונליות הנוספת הוא בכך שניתן ליצור יישום שמיועד לביצוע משימות רבות יותר. יישומים נפוצים לכך הם יישומי האינטראנט והאקסטראנט מבוססי האינטרנט. לרוב, במצבים אלה, ניתן לשער באיזה דפדפן משתמשים. דבר זה ייתן פתרון מצוין לבעיה תוך שימוש בטכנולוגיות אינטרנט.

## פתרונות מסוג EXE-based

שני סוגי הפתרונות הבאים הם אלה שדורשים פונקציונליות רבה בצד הלקוח. יישומים אלה מבוססי קבצי הפעלה (EXE), כלומר יישום מסוים יותקן במחשב הלקוח ויישאר שם.

יישום שמסתמך על האינטרנט משתמש בתערובת של תשתית מערכת ההפעלה והרשת כדי להשיג את התוכן שלו. יישומים אלה דורשים פונקציונליות של מערכת ההפעלה בשל המשימות המורכבות שמעובדות בהם. במצב זה, הדפדפן (שניתן להטמיע אותו) אינו מספק את הפונקציונליות העשירה. עם זאת, כמו בפתרון מבוסס הדפדפן, יש לחבר את היישום לאינטרנט כדי שהוא יוכל להשיג את התוכן שלו. דוגמה לסוג יישום זה היא יישום שמאחסן נתונים. כשאתה מנפה נתונים במחסן נתונים, לרוב אתה יוצר טבלאות ציר מורכבות. טבלאות אלו ניתן להציג כגרף ולהשוות ביניהן. מנתונים אלה אתה מקבל החלטה מסוימת. התהליך הגרפי והעיבוד המורכבים דורשים תת מערכת גרפית במערכת ההפעלה. עליך להתחבר לרשת משום שאינך יכול להוריד את מאגר הנתונים אל שולחן העבודה שלך. אחסון הנתונים בשלמותם דורש יכולת עיבוד גבוהה ממה שהמחשב השולחני שלך מסוגל להפיק.

rich client הוא כזה שמשתמש רק במחשב הלקוח, ואינו מודע לקיום האינטרנט. היישום יכול להיות יישום למשתמש יחיד שמתקיים בעולם משלו. עם זאת, סוגי יישומים אלה מתמעטים מפני שרוב הלקוחות מעוניינים להחליף מידע. דוגמאות ליישומי לקוח כאלה הן Word או Excel של Microsoft. יישומים אלה אינם דורשים במפורש גישה לאינטרנט, אך ניתן להשתמש בהם לצורך שיתוף נתונים.

סוג לקוח זה גורם גם לריבוי יישומים בני כלאיים. יישום בן כלאיים משתמש באינטרנט כשהוא זקוק לה, אך הוא עושה זאת בצורה נסתרת. המשתמש אינו חש בהבדל. דוגמה לכך היא Microsoft Money. יישום זה מותקן במחשב הלקוח ויכול להתקיים בבידוד מהעולם הגדול. אך כשאתה מעוניין לקבל את נתוני הבנק שלך, עליך לעבור למצב מקוון. בשלב זה מופעל דפדפן מוטמע שמתחבר לשרת הבנק שלך. הוא מוריד את הנתונים שלך ומציג אותם. בזמן התהליך כולו המשתמש אינו מקליד כתובת URL או פקודות רשת פרט לחיבור לאינטרנט. לכן, במילים אחרות, לקוח של יישום בן כלאיים עושה שימוש חלק באינטרנט.

# הטכנולוגיות השונות

כדי להכיר לך את סוגי הלקוחות השונים, אתאר את הטכנולוגיות הפנימיות :

- HTML - Hypertext Mark-Up Language, ממשק המשתמש של האינטרנט. לרוב, כשמתייחסים למונח HTML בלבד, הכוונה היא לגירסה 3.2 או 4.0. אלו גרסאות פשוטות התומכות בעיבוד מבוסס טפסים. טכנולוגיה זו היא הנפוצה ביותר מבין כל הטכנולוגיות האחרות, והשימוש בה לרוב הוא ב-thin client.
- DHTML - Dynamic HTML הוא ממשק המשתמש המתקדם יותר של דפדפני האינטרנט. DHTML שונה מ-HTML בכך שהוא תומך במודל אובייקט בשל (Full-fledged). ניתן לפנות לכל חלק בממשק המשתמש בצורה דינמית, להסיר חלקים ממנו או להוסיף לו. דבר זה הופך את הממשק לבעל עוצמה רבה. עם זאת, ישנם פחות דפדפנים התומכים בתכונה זו.
- תסריטים - ממשק התסריטים (Scripts) מוגדר במודל האובייקט DHTML. לכן, אתה עשוי לתהות מדוע כדאי להוסיף טכנולוגיית תסריטים. הסיבה לכך היא שניתן לשלוט בסוגי לקוחות רבים באמצעות תסריטים. לדוגמה, ניתן לכתוב תסריטים ב-Excel אשר יבצעו משימות מסוימות. התסריטים חלים על כל סוגי הלקוחות. היתרון העיקרי של הטכנולוגיה טמון בכך שהיא מספקת יכולת התאמה אישית של היישום, ללא צורך במיומנות מורכבת בתכנות.
- רכיבים - DHTML, תסריטים ורוב הדברים האחרים שקיימים בצד הלקוח לא היו אפשריים ללא הרכיבים (Components).
- Win32 - ה-API של Win32 הוא דרך לעבודה עם מערכת ההפעלה. טכנולוגיה זו מתאימה ללקוחות שזקוקים לגרפיקה מיוחדת או לביצועים גבוהים. ה-API של Win32 מיועד ספציפית ל-Windows. הוא אינו מהווה מודל תכנות דומה ל-HTML או ל-DHTML.

# Firewall Tier

דרגת ה-Firewall (Firewall Tier) אינה ממש דרגה, אלא התקן אבטחה חשוב שיש להזכיר אותו. דרגת ה-Firewall נמצאת בין דרגת התצוגה לדרגת הלוגיקה העסקית (Business Logic Tier). Firewall היא מערכת אבטחה המגינה על הנתונים שבתוכה מפני פריצות.

בארכיטקטורה כללית, מערכת Firewall היא רק שירות בתוך המערכת הגדולה. עם זאת, Windows DNA שונה מרוב מערכות ההפעלה בכך שהוא מעביר את נושא האבטחה לדרגת הלוגיקה העסקית (Business Logic Tier).

לדבר זה יש כמה יתרונות. הראשון מביניהם הוא שישנו מנגנון אבטחה ואימות עקבי. הבה וניקח כדוגמה טירה. תוך שימוש במנגנוני האבטחה הנוכחיים, רוכב יכול לרכב אל תוך הטירה ולהזדהות כאנונימי (Anonymous). לאחר מכן, כתלות בשירות, ייתכן ויהיה צורך באבטחת אימות. הבעיה היא שכשהרוכב נע בתוך הטירה, עליו לאמת את

עצמו כל הזמן. בנוסף, חשוב מה היה קורה אם השירות לא היה מאובטח. במצב כזה, הרוכב יכול היה להפוך למרושע ולזרוע הרס. במצב בו כל שירות מבקש זיהוי, קשה לתחזק מדיניות אבטחה עקבית. כשהרוכב נזרק מפאב אחד, הוא צריך להיזרק מכל הפאבים האחרים בטירה. בכמה מנגנוני אבטחה, זה לא קורה.

כשאתה הופך את ה-Firewall לרשות האבטחה המרכזית, כל מי שניגש לשירות כלשהו יזדהה. הוא יקבל סדרה ספציפית של זכויות והרשאות. אם ההרשאות הללו מנוצלות לרעה, ניתן לשלול אותן בצורה עקבית וקלה. בסופו של דבר, Windows DNA יוצר מערכת אבטחה פשוטה ומהירה יותר.

## Business Logic Tier

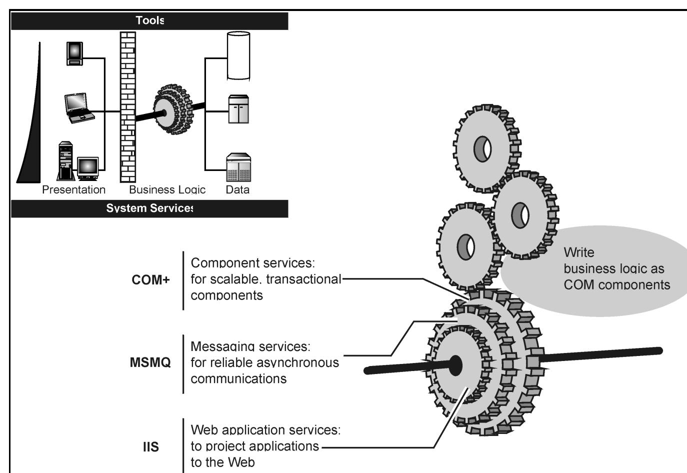
דרגת הלוגיקה העסקית (Business Logic Tier) היא לב היישום שלך. דרגה זו מעבדת ומטפלת בנתונים. התהליך העסקי (Business process) שולט ופרוצדורות שטף העבודה מטפלות בכך. דרגה זו חשובה, משום שרוב זמנך יוקדש לתכנות ועיצוב דרגה זו. בדרגה הזו, הדברים בהם תתעניין הם עיבודים מסיביים, תמיכה בתנועות (Transactions), פריסה בקנה מידה גדול, העברת הודעות (Messaging), וייתכן שגם באינטרנט. בקריאת המשפט האחרון בלבד תוכל להבין שדרגה זו מורכבת מאוד.

ל-Business Logic Tier יש שלושה שירותים עיקריים, +COM, MSMQ ו-IIS. כמובן שישנם שירותים אחרים כגון Microsoft Exchange או Microsoft SQL Server, אך הם אינם חלק פנימי של Windows 2000, כפי שמוצג בתרשים 20.4.

כל שירות מבצע משימה ספציפית. +COM אחראי על פונקציונליות התנועות וה-COM הבסיסי. IIS (Internet Information Server) אחראי על שירותי האינטרנט כגון SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol) ו-HTTP (Hyper-Text Transfer Protocol). MSMQ (Microsoft Message Queue) אחראי על עניין ההודעות. כל אחד מהשירותים הללו עשוי להשתמש בשירותים אחרים כדי ליצור שירותים נוספים. דוגמה לכך יכולה להיות תור של רכיבים. תור זה ישתמש הן ב-+COM והן ב-MSMQ כדי ליצור שירות חדש.

## COM+

הדור הבא של COM הוא +COM. דור זה מרחיב את COM על ידי הוספת שירותים שהופכים את ה-COM לשימושי עבור ארגונים. אינני טוען ש-COM אינו שימושי עבור ארגונים. COM מספק רק את הבסיס לרכיבים. כדי שרכיבי ה-COM יוכלו לבצע תנועות (Transaction), יש צורך בשירותים נוספים. זוהי מטרת +COM.




**תרשים 20.4** ארכיטקטורת Business Logic Tier

אם נעקוב אחרי ההיסטוריה של +COM ושירותי תנועות (Transaction), נגלה כי הם הראו סימני חיים בפעם הראשונה ב-SQL Server. SQL Server 6.5 הפיץ שירות שנקרא בשם DTC (Distributed Transaction Coordinator). שירות זה היה אחראי על ניהול תנועות (Transaction) ב-SQL Server. לאחר מכן, ה-DTC הופץ מחדש באריזה חדשה שנקראה בשם MTS (Microsoft Transaction Server). MTS אפשר לכתוב אובייקטי COM שביצעו תהליכים והיו מסוגלים לפקח על תנועות (Transactions). +COM מתעסק בצד התנועות (Transaction) ומשלב אותן בתוך ספריות ההפעלה של ה-COM ומערכת ההפעלה. ניתן היה להטמיע את ה-MTS אל תוך מערכת ההפעלה גם לפני כן, אך אובייקט MTS יכול לעקוף את ה-MTS. הפרצות נסתמו ב-+COM, והמערכת הפכה לעקבית ואלגנטית יותר. מנקודת מבטו של המתכנת, משמעות +COM היא שכל אובייקטי ה-COM בצד השרת יכולים לנצל את יכולת ביצוע התנועות (Transactions), כדי לתת אמינות רבה יותר ליישומים שלהם.

## שילוב TIP

+COM משתמש ב-OLE transactions כדי לנהל את סביבת עיבוד התנועות (Transactions). דבר זה נכון ספציפית עבור פלטפורמת Windows 2000 בלבד. על ידי שימוש ב-TIP (Transaction Internet Protocol), אובייקטי COM יכולים להשתתף בסביבות עיבוד תנועות (Transactions Processing - TP) אחרות. ניתן לטפל בתנועות אלו על ידי מערכת TP. דבר זה שונה מהגירסה הקודמת של MTS, בה היה צורך לטפל בתנועות על ידי ה-MTS וה-DTC.

## אבטחה משופרת

כשדנתי בדרגת ה-Firewall, הסברתי כי Windows 2000 שיפרה את מודל האבטחה. הרבה מהעבודה הזו נעשתה כחלק מסביבת +COM. בסביבת +COM, האבטחה מתבצעת בצורת אבטחה מבוססת תפקיד (Role-based) או על ידי מתן הרשאות גישה

לתהליכי העיבוד. אבטחה מבוססת תפקיד היא מנגנון בו המשתמש הוא חלק מקבוצה גדולה עם תפקיד. קבוצות התפקידים יכולות להיות מנהלים, עובדים, וכדומה. התפקידים מקלים על ניהול המשתמשים שיש להם זכויות אבטחה שונות כתלות בקבוצת המחשבים שלהם. דוגמה לכך היא רואה חשבון שיש לו גישה ברמה גבוהה לתהליכי החישוב אך גישה נמוכה לתהליכי הייצור. שלוחה אחת של האבטחה מבוססת התפקיד ב-+COM היא היכולת להחיל את האבטחה ברמת השיטה עבור אובייקטי ה-COM.

## ניהול מערכת מרכזית

במהדורה הקודמת של Windows NT, הטיפול ברכיבי COM התבצע תוך שימוש בסייר ה-MTS. ניתן היה לנהל הגדרות DCOM תוך שימוש ב-DCOMCNFG. ב-Windows 2000, שני כלים אלה הוחלפו ב-Component Services Explorer. כלי זה משלב ומרחיב את יכולת ניהול אובייקטי +COM. המשימות שבו כוללות פריסה, ניהול וניטור יישום +COM.

## בסיסי נתונים שוכני זיכרון (In-Memory Databases - DB)

אחד מהשירותים החדשים של Windows 2000 הוא IMDB. IMDB הוא בסיס נתונים השוכן בזיכרון ושומר טבלאות בזיכרון RAM. כיום, יישומים רבים דורשים ביצועים טובים. ביישומים אלה, רוב הנתונים מיועדים לקריאה. דוגמאות לכך כוללות קטלוג מוצר. טבלה זו משתנה, אך באיטיות. רוב הזמן הגישה אליה היא לצורך קריאה, וניתן לאחסן אותה במטמון. כאן ה-IMDB יעיל ביותר. בעזרת השירות, אתה יכול להכניס את טבלת קטלוג המוצר במטמון שהגישה אליו מהירה. IMDB אינו שירות כללי. הוא משרת מטרה ספציפית ופונקציונליות שנקבעה על ידי המתכנת.

## רכיבים בתור

ב-DCOM (Distributed COM) משתמשים כך שניתן יהיה לקרוא לאובייקט COM בין שני מחשבים שונים. DCOM הוא פרוטוקול סינכרוני. כשמתקבלת קריאה ממחשב אחד, הנמען חייב להיות זמין. אם הנמען אינו זמין, הקריאה תיכשל, וכל פעולה שקשורה בה תיכשל אף היא. אם פעולה זו מבוצעת בתוך הקשר של תנועה (Transaction), תנועה זו תיכשל אף היא. במצב זה, עליך לשלב מנגנון לקריאה חוזרת במקרה של כישלון.

ב-Windows 2000, ישנו פתרון קל יותר בדמות תור רכיבים. תור רכיבים מאפשר לקרוא לרכיב אחר במחשב אחר בצורה אסינכרונית. תורות רכיבים משתמשים ב-MSMQ כבסיס ארכיטקטורת ההודעות, ולפיכך אם הקשר או הקריאה נכשלים, ניתן לחזור על הקריאה עד לביצוע קריאה מוצלחת.

## שירותי אירועים

ב-Windows NT 4.x, אין אפשרות לחבר בקלות אירועים לרכיבים. ישנה טכנולוגיה בשם COM Connection Points, אך היא מוגבלת מאוד. מה שנדרש לשם כך היא טכנולוגיה שתאפשר זוגות אירועים חופשיים, בהם מקבל האירוע יכול להיות פעיל או לא פעיל.

מנגנון אירועים זה נקרא אירועי COM+. על ידי שימוש באירועי COM+, ניתן לבצע קריאות יחידות או מרובות. מקבל האירוע מתחבר ללקוח תוך שימוש בטכניקת רישום.

מנגנון ההפצה והרישום הוא כזה שבו המפיץ (publisher) מודיע לשירותי אירועי COM+ שיש לו מידע להפצה. נמען שמעוניין לקבל מידע זה מציין זאת בפני שירות אירוע COM+ על ידי רישום. כשלמפרסם יש נתונים שהשתנו, יעבירו שירותי אירועי COM+ את השינויים למנויים השונים.

## איזון עומס דינמי

לאחר ש-Windows 2000 יצאה לאור, ניתן היה לאזן דינמית עומס של אובייקט COM+. איזון עומס דינמי (Dynamic Load Balancing) הוא טכניקה לחלוקה שווה של עומס העיבוד על סדרת מחשבים. איזון עומס הוא תהליך שרת והוא שקוף ללקוח שמבצע את הקריאה לשרת.

## שירותי אינטרנט

ישנם מיגוון שירותי אינטרנט המסופקים על ידי Internet Information Server (IIS). שירותים אלה כוללים:

- HTTP - זהו פרוטוקול האינטרנט והשימוש בו הוא בדפי אינטרנט או בכל תוכן שדפדפן האינטרנט דורש.
- FTP - זהו הפרוטוקול בו משתמשים באינטרנט כדי להעביר קבצים. הוא מצויד במיוחד עבור תנועות קבצים ולפיכך, הוא יעיל יותר.
- SMTP - זהו הפרוטוקול בו משתמשים כדי להעביר הודעות דואר אלקטרוני ממחשב אחד למשנהו. ניתן להשתמש בו יחד עם POP.

הדבר שמייחד את ה-IIS יחסית לשרתי אינטרנט אחרים הוא השילוב שלו עם תנועות (Transactions). כשמתבצעת דרישה כלפי ה-IIS, זוהי דרישת תנועה (Transaction). על ידי שימוש בתנועות כבסיס לטיפול בדרישות, ניתן להשתמש בכל יתרונות ה-+COM. דבר זה הופך את COM+ גם לגמיש יותר כשהוא זקוק לאיזון עומס דינמי או משופר.

כדי לבנות תוכן עבור ה-IIS, ניתן להשתמש ב-ISAPI (Internet Server API) או לרוב בדפי ASP (Active Server Pages). טכנולוגיית ASP היא סביבת תסריטים בה הלוגיקה מתבצעת מתוך תסריטים. שפת התסריט נייטרלית ודינמית לחלוטין. השימוש בה נעשה תוך שילוב עם אובייקט COM+ כדי לספק יישום רב תכונות. ניתן להשתמש ב-ASP כדי ליצור DHTML או XML. למרות שזה אינו חלק מ-Windows 2000, ניתן לטפל בתהליך יצירת יישומי ASP, DHTML ו-XML תוך שימוש ב-Visual InterDev.

## שירותי הודעות

הודעות אסינכרוניות (Asyncronous messages) מטופלות תוך שימוש ב-MSMQ (Microsoft Message Queue Server). שיטת ההודעות (Messaging) היא דרך שונה לחלוטין לכתיבת קוד. שיטה זו מניחה שלא קיים קשר בין הלקוח לשרת. היא מניחה כי בסופו של דבר הפעולות יבוצעו. מודל ההנחות המאוחרות מאפשר לכתוב יישומים שמתפקדים ברשתות לא אמינות או בתנאים לא אמינים.

MSMQ היא ארכיטקטורה המונעת על ידי אירועים. הנתונים נדחפים לשרת, והשרת חייב להגיב לאירוע שנקרא. שיטה זו אינה חדשה משום ש-Windows עצמה פועלת על פיה.

## פעולה הדדית: מערכות, יישומים ונתונים

משימה אחת היתה להכין את Windows 2000 לאינטרנט. אך ישנה משימה גדולה אחרת, והיא להכין את מערכת ההפעלה לשילוב מערכות מיושנות (Legacy Systems). מערכות אלו קיימות מפני שהן עדיין מבצעות את עבודתן כהלכה. לפיכך, לא כדאי להחליף אותן, אלא לשלב אותן בארכיטקטורה הכללית.

Windows 2000 מספקת יכולת לשלב מיגוון טכנולוגיות מיושנות בדרכים הבאות:

- שילוב MSMQ - שילוב הודעות מסופק עבור MSMQ תוך שימוש במיגוון מוצרים. כדי לחבר בין פלטפורמת Windows למוצר MQSeries, ניתן להשתמש בשרת SNA. כדי להתחבר ל-MSMQ מפלטפורמות אחרות (UNIX, MVS, OS/2, ואחרות), ניתן להשתמש בלקוח FalconMQ מבית היוצר Level8.
- COM Transaction Integrator - על ידי שימוש בממשק זה, ניתן להרכיב תנועות (Transaction) מסוג CICS ו-IMS כסדרה של אובייקטי COM. ב- COM Transaction Integrator, קיימת סדרה של כלי פיתוח ולוגיקה עסקית (Business logic) אשר מכסים את התנועות של IBM Mainframe. התקשורת לתנועות למחשב IBM מטופלות על ידי Windows SNA Server.
- Universal Data Access - שכבה זו עדיין לא תוארה, אך היא שכבת הגישה לבסיס הנתונים. על ידי שימוש במנהלי התקנים (Drivers) טבעיים, ניתן להתחבר למשאבים שנמצאים בצד הפלטפורמה האחרת.
- DCOM על גבי UNIX/Mainframe - זוהי טכנולוגיה שמאפשרת לכתוב אובייקטי COM בפלטפורמות השונות מ-Windows. Software AG סיפקה את הממשק הראשון עבור Solaris. ממשק זה הורחב והוא כולל את פלטפורמת ה-Mainframe. כיום, הוא הורחב אף יותר והוא כולל ארכיטקטורה מלאה הנקראת בשם EntireX iNTegrator.
- TIP/XA-Transaction Integration - כאן נמצאים כמה תקנים של פרוטוקולי תנועות (Transactions). TIP כבר תואר קודם לכן. תקן אחר הוא XA-Protocol. השילוב המסופק נעשה דרך מנהל/מחלק המשאבים.

## כמה מילות סיכום

דרגת הלוגיקה העסקית (Business Logic Tier) היא דרגה חשובה מאוד. היא מכילה את המספר הגדול ביותר של אובייקטי COM אשר מספקים שירותים ספציפיים. גמישות זו דרושה כדי לבנות יישומים ארגוניים ברמה בינלאומית אשר מסוגלים לקבל, לעבד ולשלוח בחזרה את המידע מדרגת הנתונים.

# Data Tier

הדרגה האחרונה היא דרגת הנתונים (Data Tier). זוהי דרגת ניהול המשאבים. בניגוד לדרגת הלוגיקה העסקית (Business Logic Tier), אין כאן עיבוד נתונים. לחילופין, מוגדרת כאן משימת הטיפול בכמות הנתונים העצומה. בסיסי נתונים ומשאבים גדלים ומתנפחים בכל יום, ומשימה זו הופכת לקשה יותר ויותר.

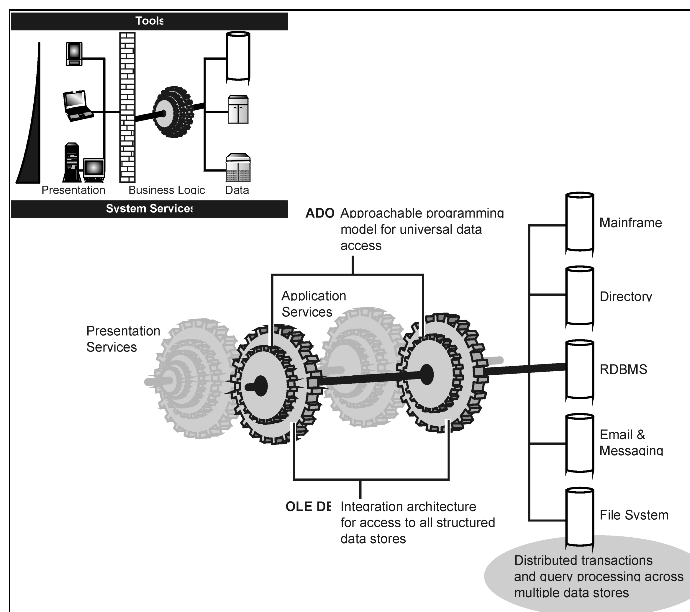
אסטרטגיית UDA (Universal Data Format) שונה במקצת מהיצרנים האחרים. UDA מאחסנת ומטפלת בנתונים. חלק מהנתונים נמצאים בגיליון אלקטרוני, חלקם במעבד תמלילים, וחלק אחר בבסיס נתונים יחסי. דרך אחת לטפל בנתונים אלה היא ליצור מאגר אחסון שמסוגל לטפל בכל סוגי הנתונים הללו. במציאות היומיומית, מאוחסנים הנתונים בבועה (Blob).

הבועה מטפלת רק בנתונים כמו יישות בינארית. היא אינה מבטיחה את היחסים ביניהם. לדוגמה, אם הבועה המאוחסנת מתייחסת למסמך אחר, לא תתגלה התייחסות לא תקפה. בדרגת הנתונים (Data Tier), להתייחסות לא תקפה זו אין משמעות. עם זאת, ברמת הלוגיקה העסקית (Business Logic Tier) יש לה משמעות רבה. היא יכולה לגרום לביטול תנועה (Transaction) או מספר תנועות (Transactions).

הפתרון לכך הוא להבטיח שדרגת הלוגיקה העסקית (Business Logic Tier) כותבת את הבועה (Blob) בצורה נכונה. אך דבר זה מוסיף שלבי תכנות שאמורים להיות חלק ממשאבי הנתונים. טיפול בהתייחסויות הוא פעולת נתונים. כדי להוסיף לטיפול בהתייחסויות, יש להוסיף מודול חיצוני למאגר הנתונים, וזוהי אינה הדרך הטובה ביותר לעשות זאת.

הפתרון הטוב יותר הוא להניח שכל משאב מטפל בפורמט הנתונים שלו. אך המשאב חייב להתאים למודל אובייקט נפוץ. דבר זה מאפשר לטפל, להעתיק ולהתייחס לנתונים בצורה שאינה תלויה בפורמט.

מודל האובייקט UDA נקרא מודל אובייקט OLE DB (ראה תרשים 20.5). זהו פורמט ברמה בסיסית (Low level) אשר כולל את מפרט תהליך הגדרת התנועות (Transactions) וה-Rowset. אובייקט OLE DB פועל במהירות גבוהה וברמה הבסיסית, ומשמעות הדבר היא שכתיבה ל-OLE DB מצריכה איסוף נתונים. למרות שזהו אינו חלק מחבילת Windows 2000, Visual C++ מכיל שכבת תבנית קטנה שנקראת OLE DB Consumer Templates. שכבה זו מפשטת את OLE DB בלי להאט אותה או לפגוע ביעילותה.




**תרשים 20.5** Universal Data Access (UDA)

היתרון במודל האובייקט OLE DB הוא שהמודל אינו מוגבל. יש להטמיע סדרת משנה ספציפית. אין דבר שיעצור יצרן כלשהו מלהרחיב את מודל האובייקט כך שהוא יספק גישה מכוונת לנתונים שלו. שוב, דבר זה דומה לדרגת התצוגה בה יכולת ההגעה למקומות רבים באה על חשבון פונקציונליות עשירה.

ה-OLE DB שייך מטבעו לרמה הבסיסית, ולכן Visual Basic או תסריטים אינם יכולים להשתמש בו ישירות. לשם כך, נוצר מודל מפושט בשם ADO (Active Data Objects). מודל אובייקט זה קטן מאוד אך רב עוצמה והוא מאפשר גישה לרוב תכונות ה-OLE DB. עדיין נותרה שאלה לא פתורה: כיצד ה-OLE DB מגיע ל-Blob? התשובה היא שהוא עושה זאת כמו אובייקטי COM אחרים, אך דבר זה תלוי ביישום של ספק ה-OLE DB. בשל כך, ה-ADO אינו צריך להיות גדול כדי לטפל בכל הפורמטים של הנתונים השונים. האחריות ליצירת אובייקטי COM נופלת על הספק.

# שילוב ASP בחזון Windows DNA

השאלה שנשאלת כעת היא כיצד אתה, כמתכנת ASP, תשתלב לתוך חזון Windows DNA? ASP הוא סביבת תסריטים שמופעלת בצד השרת. לפיכך, חלק מהתשובה הוא שהתוכנית שלך תשכון בדרגת הלוגיקה העסקית (Business Logic Tier). אך זוהי אינה הדרגה היחידה, משום ש-ASP מייצר קוד GUI. משמעות הדבר היא ש-ASP אחראי בצורה עקיפה על דרגת התצוגה. לרוב, ASP ייצור קוד HTML. במקרה

של thin client ,ASP הוא דרגת התצוגה. אך במקרים בהם נעשה שימוש ב-DHTML או בלקוח בן כלאיים (Hybrid Client), הוא יוצר את קוד הממשק שייצור לוגיקת הפעלה בדרגת התצוגה. במקרה כזה, עומס העיבוד מתחלק בין דרגת התצוגה לדרגת הלוגיקה העסקית (Business Logic Tier). בכל המצבים הללו, אנו מניחים שהלקוח מתבסס על האינטרנט.

# בניית יישומי אינטרנט

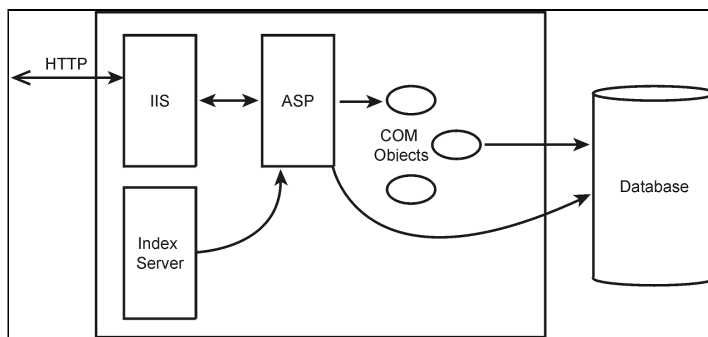
התהליך שהוגדר זה עתה נקרא בשם **בניית יישומי אינטרנט**. יישומי אינטרנט (Web Applications) שונים מאתרי אינטרנט שפועלים ב-HTML בכך שהם מבצעים לוגיקה עסקית (Business logic). זהו הבדל חשוב משום שאתרי אינטרנט אינפורמטיביים עשויים להשתמש ב-ASP אך הם מפיקים יותר נתונים. דוגמאות לכך כוללות חיפוש בקטלוג מקוון. ניתן להשתמש ב-ASP כדי לבצע את החיפוש, אך הלקוח מעוניין בנתונים.

ביישומי אינטרנט, ישנו תהליך עסקי (Business process) שיש לפתור. התהליך העסקי עשוי להיות רישום הלקוח לוועידה. הוא עשוי לכלול עיבוד משכנתא בצורה מקוונת. יישום אינטרנט מכיל לרוב כמה שלבים, והוא מעבד ומשנה את הנתונים. דבר זה מנוגד לאתרי אינטרנט אינפורמטיביים בהם הנתונים הם ברובם לקריאה בלבד.

ביישום אינטרנט, אין אמון בלקוח. לשרת לא אכפת מהיכן הלקוח מתקשר. השרת שולח את כל תוכן ה-HTML הדרוש ללקוח בזמן שהלקוח דורש אותו. הדרישה היחידה היא שללקוח יהיה יישום הפעלה שיעבד את התוכן המתקבל. לרוב, יישום ההפעלה יהיה דפדפן HTML.

היישום שלך מניח שכל השלבים הקריטיים קורים בשרת. כל המצבים הנוכחיים נשמרים בשרת. במקרה של thin client, הלקוח חייב להישאר מחובר לרשת במשך ביצוע המשימה כולה. עם זאת, היתרון בארכיטקטורה הוא שכשמופיעה גרסת יישום חדשה, השרת הוא הצד היחיד שצריך להתעדכן.

ארכיטקטורת יישום האינטרנט מתוארת בתרשים 20.6. שרת IIS אחראי על הטמנת דרישת HTTP מלקוח מסוים. דרישה זו מומרת מ-HTTP ל-ASP. דרישת ASP היא תסריט שיש לו גישה לסביבת COM. בחלק השני של הספר, מתוארים אובייקטי COM ברי התקנה של ASP. החלק הרביעי, מציג כיצד ליצור אובייקטי COM מותאמים אישית. בכל אחד מהמקרים, יש לתסריט גישה לפונקציונליות מסוג שירות. שירות הוא דבר שונה משום שהוא מבצע פעולה אחת. בחלק השני, הדוגמה שניתנה היא רכיב SMTP. ASP יכול להשתמש בה כדי לשלוח נתוני משוב אודות פעולה שהלקוח ביצע.




**תרשים 20.6** ארכיטקטורת יישום אינטרנט

# אחזור נתונים והכנת אינדקס

למרות שכבר היה ניתן לעשות זאת בעבר, ב-ASP 2.0 ובגרסאות עתידיות של Index Server קל יותר ליצור אינדקס של נתוני יישום אינטרנט. דבר זה חשוב משום שמשתמש הקצה עשוי לרצות לבצע שאילתה באתר אינטרנט עבור יישום ספציפי. היכולות והפעולה ההדדית של ASP עם Index Server מתוארות בפרק 15.

# שירותי תנועות

כעת, נרחיב את הטווח ונתייחס לצרכי יישומי האינטרנט. ציינתי כי יישום Windows DNA פותר תהליכים עסקיים (Bussiness processes). על תהליכים אלה להיות מהירים, אמינים וברי הרחבה. הרכיבים המתוארים עד פה אינם פותרים אף אחת מהבעיות הללו.

COM+ פותר זאת על ידי הפיכת דף ASP לחלק מתנועה (Transaction). לאחר שדף ASP הפך להיות חלק מתנועה, הוא משפר את תפעול העסק (Business operation). דוגמאות לכך כוללות איזון עומס עבור ביצועים וביטול תנועה (Transaction), לשם האמינות והעקביות.

מדוע יש צורך בכך? חשוב על הדוגמה בה יישום אינטרנט ניגש לשני בסיסי נתונים השוכנים בשני מחשבים שונים. התהליך העסקי (Business process) הוא העברת כסף מחשבון אחד למשנהו. בבסיס הנתונים הראשון, החשבון מחויב בסכום X, ואילו החשבון השני מזוכה בסכום X.

בעולם האמיתי, אחת הפעולות האלו עשויה להיכשל. אם זה קורה, כמובן שגם השנייה חייבת להיכשל. אחרת נתוני המאזן יהיו שגויים, והכסף יוכפל או יאבד. בשלב זה, תוכל, בעזרת הידע שלך אודות בסיסי נתונים ב-ASP, לדעת איזה תנועה (Transaction) נכשלה ולבטל את התנועה האחרת. דבר זה עשוי להיות מייגע משום שעליך לברר אם פעולת הביטול הצליחה.

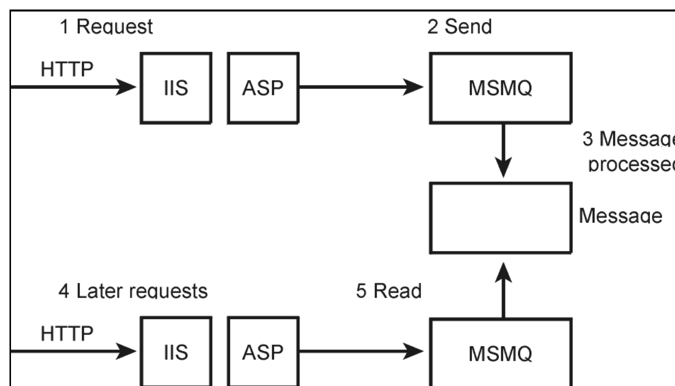
על ידי שימוש בתנועות (Transactions) ניתן לבצע פעולת ביטול ובדיקה עקבית זו בצורה אוטומטית. +COM מצליח לבצע זאת על ידי הפיכת דף ה-ASP שלך וכל אובייקט COM אליו מתבצעת גישה לחלק מתנועה (Transaction). אם כל תסריט או אובייקט COM מבטל את התנועה, כל בסיסי הנתונים המעורבים בה יבטלו אותה גם הם. בצורה זו, הרבה יותר קל לנקות את שאריות התנועה המבוטלת.

# שירותי הודעות

השירות העיקרי השני אשר נדרש כדי לבנות יישומים מסוג Windows DNA הוא העברת ההודעות. יישום האינטרנט של ASP הוא תהליך סינכרוני. עליך לבצע דרישה לדף אינטרנט, ולהמתין עד שה-ASP סיים לעבד את הדף. אם התהליך הזה לוקח זמן רב, הדפדפן ימתין זמן רב. צורת פעולה זו אינה רצויה משום שהדפדפן עלול ליצור שגיאת Timeout ולציין בכך שאין אפשרות למלא את הדרישה.

בתרשים 20.7, דוגמת שטף העברת ההודעות תהיה בצורה הבאה:

1. בשרת ה-IIS מתקבלת דרישת HTTP. דרישה זו גורמת להפעלת תסריט ASP ועיבוד מסוים. בזמן העיבוד, עשויה להתבצע קריאה לתסריטים אחרים או לשירותי/רכיבי COM אחרים. כל זאת עשוי להתבצע בהקשר של עסקת +COM.




**תרשים 20.7** ארכיטקטורת העברת ההודעות

2. בנקודה מסוימת בתסריט ASP, רכיב MSMQ נקרא ונוצרת הודעה. ההודעה אינה מעובדת. היא רק נוצרת ומועלית לשירות MSMQ. תסריט ASP ממשיך את העיבוד והוא שולח הודעה בחזרה ללקוח ממנו ההודעה נשלחה. בנקודה זו, הלקוח ושירות ה-MSMQ פונים לשני נתיבי הפעלה שונים. הם אינם מסתמכים אחד על השני לקבלת נתונים או מצבים ספציפיים.

3. שירות MSMQ שולח את ההודעה ליעדה. בנקודה זו, MSMQ מילא את תפקידו. תהליך עיבוד שנמצא בהאזנה ביעד מעבד את ההודעה. סביר שתהליך העיבוד ביעד שולח הודעת תוצאת עיבוד בחזרה ל-MSMQ. MSMQ מבצע שוב את תפקידו ושולח את ההודעה מהיעד בחזרה למקור.

4. בנקודת זמן אחרת, הלקוח יוצר דרישה נוספת לשרת ה-IIS. דרישה זו מפעילה מחדש תהליך עיבוד ASP.

5. כעת, במקום ליצור הודעה, מתבצע חיפוש אחרי הודעה מעובדת ותוכנה מוחזר ללקוח.

בזמן הפעלת יישומי MSMQ, שים לב לכך שהלקוח חייב ליזום את הדרישה. אין אפשרות לקריאה חוזרת (CallBack) בשם שרת ה-IIS. תהליך זה דומה לשיטת השאילתה (Polling), אך היתרון שבו הוא שהעיבוד יכול להתחיל וניתן לבצע משימות אחרות בין השאילתות השונות. דוגמה לשימוש בהודעות היא קבלת בקשה למשכנתא. לרוב, המשכנתאות אינן מאושרות בטווח של שניות. יש לבדוק את הנדל״ן, למלא את פרטי הביטוח, ולבצע דברים נוספים לפני שיתקבל אישור.

# סיכום

Windows DNA - ארכיטקטורה לבניית יישומים עסקיים (Business applications). אתה מתמקד בהיבט העסקי (Business) ומשאיר למערכת ההפעלה לעסוק בקוד המערכת ובבסיס הנתונים. בעזרת Windows DNA, ניתן גם לתכנן מערכת משום שהשירותים השונים כבר זמינים. התפקיד שלך כמתכנת ASP הוא לכתוב קוד תסריטים שניגש לרכיבי COM, אשר מתבצע בתוך תנועות (Transactions). אתה עשוי לכתוב כמה רכיבי COM, אך ההתמקדות היא בתסריטי ASP. בתהליכים אסינכרוניים, תשתמש גם ב-MSMQ כדי לבצע פעולות אצווה (Batch).

בפרקים הבאים תיישם את הידע אותו צברת עד עתה בבניית יישומי אינטרנט. פרקים אלה יתמקדו בדרך השימוש בתנועות (Transactions) ובשירותי הודעות (Messaging services).

## פרק 21

# שימוש ב-ASP עם MTS
# (Microsoft Transaction Server)

**בפרק זה:**

- **מבוא ל-Microsoft Transaction Server**
- **MTS ורכיבים**
- **שימוש ב-MTS להגדלת יכולת ההרחבה והביצועים של ASP**

רוב אתרי האינטרנט תלויים בחסדי הסביבה שלהם. כמה פעמים ביקרת באתר אינטרנט שלא תיפקד עקב בסיס נתונים לא פעיל? תקלת תכנות? פיגור קיצוני? שרת עמוס? כיום, יותר ויותר יישומי אינטרנט מורכבים מסתמכים ברובם על עבודה עם כמה מערכות שונות לביצוע פעולה. פעמים רבות, אם מערכת אחת אינה זמינה, הפעולה לא תושלם כראוי. במצב מעין זה, תישאר עם פעולה גמורה למחצה, או עם אתר אינטרנט מת. בפרק זה, נדון ב-MTS (Microsoft Transaction Server) ובדרך בו הוא יכול לסייע לך להימנע מתופעת אתר האינטרנט המת (Dead Web Site) על ידי בדיקת יכולת השדרוג שלו, והסבילות שלו לנפילות ואמינות.

# מבוא ל-<br>Microsoft Transaction Server

בפרק 20 סקרת את הגישה והרעיון העומדים מאחורי ארכיטקטורת **D**istributed interNetwork **A**pplication של Windows. חלק גדול מחידת ה-DNA נפתר על ידי Microsoft Transaction Server (MTS). MTS, על פי שמו בלבד, הוא תעלומה. אם היית צריך לבסס את הידע שלך בנוגע ל-MTS על שמו בלבד, היית עשוי להגיע למסקנה שהוא קשור לתנועות (Transactions) ועיבוד תנועות, ולהאמין בזאת. לרוע המזל, השם הגרוע שנבחר ממעיט בעוצמ, וריבוי התכונות השוכנות בלב השרת.

בפרק זה, תגלה שאולי היה עדיף לתת ל-MTS שם אחר, כגון Microsoft Components Server או Microsoft Component Manager. למרות שהתנועות הן חלק גדול מפונקציונליות MTS, מה ש-MTS מספק למפתח הוא בסיס רב עוצמה לניהול יישומים ברי הרחבה. הבה ונחקור את MTS לעומק, כדי לדעת מה משמעות הדבר.

## תפקיד MTS

בדרך כלל, כשאתה עובד עם יישומים מבוזרים (ובמיוחד אמור הדבר לגבי יישומי אינטרנט), אתה מסתמך על סדרת דפי אינטרנט שעושים את העבודה (קרוב לוודאי שמדובר בדפי ASP), ועל רכיבים לוגיים שמפותחים בהתאמה אישית. כדי להשתמש ברכיבים אלה, דפי האינטרנט שלך חייבים לקרוא לרכיבים אלה על בסיס צורך כלשהו. הרכיבים שלך, מצידם, חייבים לנהל את עצמם ולהבטיח שהביצועים לא יונחתו על ידי קריאות חוזרות.

Microsoft Transaction Server מספק לך, כמפתח, סדרת תכונות שמשתלטות על חלק גדול מהעבודה המייגעת הקשורה בניהול יישומים מבוזרים. ספציפית, MTS קובע כיצד היישום שלך מתקשר ופועל יחד עם רכיבים אחרים. בחלק הרכיבים של הספר, למדת אודות COM (Component Object Model) וכיצד להשתמש ברכיבי COM ביישום האינטרנט שלך.

## JIT-ו Component Instances

כשאתה עובד עם יישום אינטרנט, השימוש ברכיבים גורר אחריו סדרת שלבים צפויים:

1. **1.** דף ה-ASP יוצר מופע (instance) חדש של הרכיב כאובייקט באמצעות פעולה המכונה יצירת מופע (Instantiation). לדוגמה, על ידי ביצוע הקריאה:

```
Set <variable> = Server.CreateObject("<Component>")
```

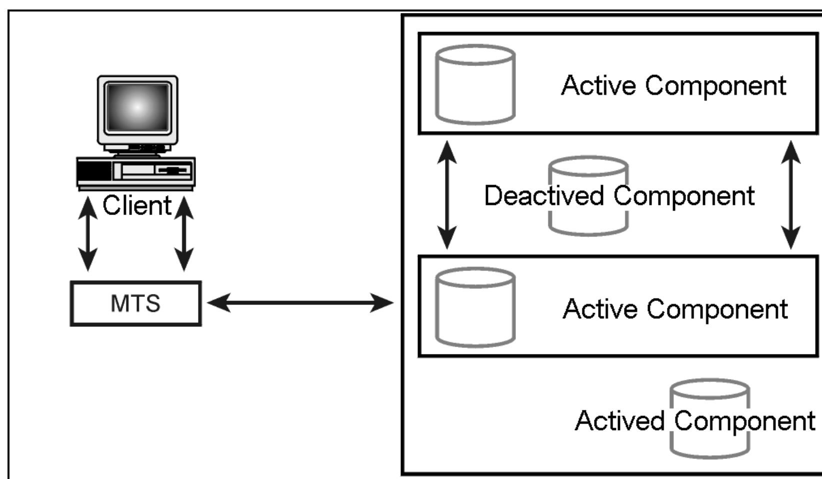
2. **2.** מופע (instance) האובייקט/רכיב מאותחל לשימוש הראשון.
3. **3.** דף ה-ASP קורא לפונקציונליות הרכיב על ידי התסריט שבקובץ.
4. **4.** לאחר שדף ה-ASP סיים להשתמש ברכיב, נמחק מופע (instance) האובייקט מהזיכרון. דבר זה נעשה על ידי הוראה ספציפית (לדוגמה, Set <variable> = Nothing), או על ידי הפגת תוקף המשתנה לאחר השלמת העיבוד.

עם זאת, גישה זו זרועה בבעיות. בעיית המפתח בגישה היא השפעתה על ביצועי יישום האינטרנט שלך. קח בחשבון שבכל פעם שיישום אינטרנט יוצר מופע (instance) של רכיב עבור משתמש, רכיב זה מוקדש לאותו המשתמש. בנוסף, קח בחשבון שלמופע (instance) יש אורך חיים קטן (לרוב, דף אחד). דבר זה מוסיף תקורה רבה, במונחים של זמן עיבוד וזיכרון עבור כל משתמש שפועל עם יישום האינטרנט שלך. לדוגמה, אם דף ASP יחיד ביישום שלך קורא לרכיב פעם אחת בלבד, משמעות הדבר שיש ליצור מופע (instance) של הרכיב בזיכרון עבור כל משתמש סימולטני, ולמחוק אותו לאחר השימוש. זו אינה צורת עבודה יעילה.

הדברים עלולים להסתבך אף יותר כשאתה מערב מחשוב מבוזר. לדוגמה, מה יקרה אם הרכיב שלך יספק פונקציונליות אחורית כגון database connection? יהיה עליך ליצור חיבור (connection) לבסיס הנתונים עבור כל מופע (instance) של הרכיב. אם יש לך 30 משתמשים בו-זמנית שניגשים ליישום, הם עלולים לגרום ליצירת 30 חיבורים שונים לבסיס הנתונים (database connections). כעת, אין לך רק את תקורת העומס של יצירת ומחיקת היישום, אלא גם תקורת עומס ביצירת session בבסיס הנתונים. דבר זה משפיע לא רק על הביצועים, אלא גם על העלות - עליך להתחשב בדרישות הרישוי עבור כל session פתוח בו-זמנית. למרות שניתן לטפל בכך על ידי עיצוב קפדני, תכנון ויחס נכון של מאגר חיבורים (connection pooling) שמסופקים על ידי מערכת ההפעלה ושרתי בסיסי הנתונים, הבעיה נשארת בעינה.

## Pooling Instances

מה הפתרון לכך? אם ניחשת שהפתרון הוא צירוף מופעי רכיבים (Pooling Instances), גילית את הסודות הפנימיים של MTS. Microsoft Transaction Server עוקף את הבעיות הללו על ידי יצירת תוכן מתמיד עבור הרכיבים שלך. תוכן זה מספק ליישום שלך מאגר מופעים (instances) עבור הרכיבים שלך. ניתן לשתף מופעים (instances) אלה בין משתמשים ואפילו בין יישומים, תוך הקטנת הצורך ביצירת מופע (instance) רכיב עבור כל משתמש. רעיון זה מודגם בתרשים 21.1.



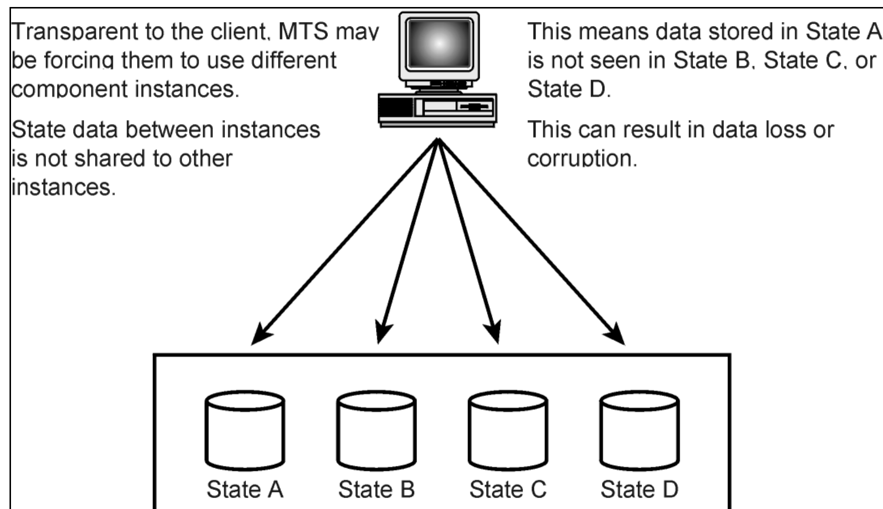

**תרשים 21.1** היישומים מתייחסים לאובייקט הקשר, שמתייחס בתורו לעותקים בתוך MTS

לאחר שיישום סיים להשתמש ברכיב, הוא יכול לשחרר את תוכן האובייקט שלו, שבתורו ישחרר את מופע (instance) הרכיב בתוך MTS. MTS מטפל בכמה מופעים (instances) בו-זמנית, ולמעשה הוא אינו מסוגל להסיר את המופע (instance) שבתוך MTS מיידית, משום שמשתמש או יישום אחר עשויים להשתמש בו. עיקרון המפתח אותו יש להבין כאן הוא שהיישום שלך אינו מתייחס ישירות למופע הרכיב. לחילופין, הוא מתייחס לתוכן האובייקט שבתוך MTS. MTS קובע בעצמו איזה מופע (instance) של הרכיב זמין לשימוש או אם יש ליצור מופע חדש כדי לענות על דרישות תוכן האובייקט.

## הבנת מצבים (State) והקשר (Context)

רכיבים ש-MTS מטפל בהם נהנים מתכונות אחרות שהיו דורשות תכנות מורכב במצבים אחרים. תכונה אחת חשובה של MTS היא המצב (State). המצב וצורת ההתנהגות קשורים אחד בשני. התנהגות הרכיב עשויה להיות עקבית לאורך כל חייו (מזמן ההפעלה שלו ועד לזמן המחיקה). אתה יכול להתייחס להתנהגות כקוד הפנימי או כפעולות שנעשות מאחורי הרכיב. תכונת הנדיפות של הרכיבים נובעת מתוך **מצב**. המצב מתייחס לנתונים או לטבע הלא קבוע של הרכיב. המצב יכול לציין בעיות מסוימות כשאתה עובד עם רכיבים תחת הנחיית MTS.

היישום שלך לא יעבוד תמיד אם אותו מופע (instance) של הרכיב בכל פעם שתוכן האובייקט נקרא, ולכן נתוני המצב עשויים שלא להיות זמינים, או גרוע מכך - אתה עשוי למצוא נתוני מצב שונים. תפישה זו מוצגת בתרשים 21.2.




**תרשים 21.2** למרות שהיישום שלך מאמין שהוא מקושר לאותו מופע (instance) הרכיב בכל פעם, לרוב, למעשה, המצב אינו כזה

במצב האידיאלי, הרכיבים שלך הם חסרי מצב, דבר שמאפשר ל-MTS למחוק וליצור מחדש רכיבים על פי הצורך. למרות שהמצב הזה אידיאלי, קיים לעיתים הצורך ליצור רכיבים שזוכרים נתונים בין קריאות ופעולות שהם ביצעו עם היישום שלך. כאן נכנסים לתמונה MTS והמצבים. MTS מספק למפתחים ארבעה מאגרים שמיועדים לנתוני מצב, אותם ניתן לנהל בצורה מבוקרת. ארבע השיטות הן:

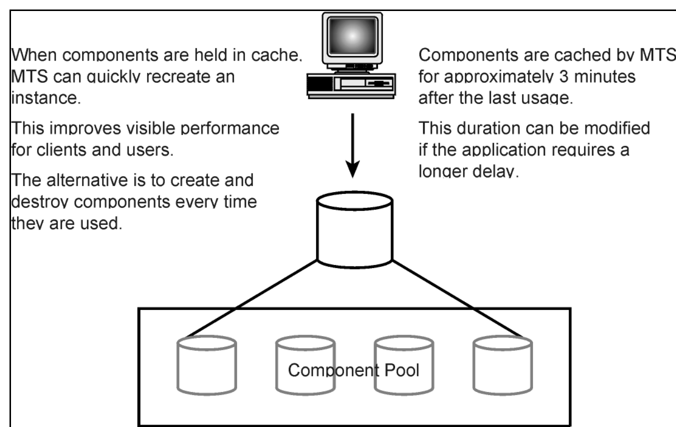
- מאגר מצב בצד הלקוח (Client State Storage) - כל נתוני המצב מאוחסנים ביישום הלקוח, ובכך מאפשרים לרכיב להתנהג בצורה שדומה מאוד לאובייקט חסר מצב. כל פעילות שנעשית עם הרכיב כוללת החלפת נתוני מצב בין MTS ליישום הלקוח.
- מאגר מצב בצד האובייקט (Object State Storage) - כשאתה לוקח את הנתיב המסורתי של אחסון נתונים בתוך רכיב, אתה עובד בשיטת אחסון מצב בצד האובייקט ב-MTS. עם זאת, יש להשתמש בשיטה זו בזהירות, משום שהיא מונעת מ-MTS לטפל ברכיבים שלך (הפעלה וביטול Just-in-time). כש-MTS מוחק את הרכיב, כל נתוני המצב שהאובייקט הכיל אובדים. דבר זה שווה ערך להסרת ההתייחסות לאובייקט על ידי קביעת ערכו כריק. רכיבים אלה גם אינם מתרחבים בצורה טובה, ולכן יש להימנע משימוש בהם במידת האפשר.
- מאגר מצב בצד בסיס הנתונים (Database State Storage) - במציאות, רוב המערכות מסתמכות על בסיס נתונים כמאגר נתונים קבוע. רכיבים שמשתמשים במאגר מצב בצד בסיס הנתונים מתבססים על קישור OLE DB או ODBC למקור

נתונים כדי לאחסן נתוני מצב של רכיב, ובכך הם מאפשרים לו לקבל ולאחסן נתונים כפי שנדרש. נתיב זה אידיאלי למצב בו אתה אינך מעוניין להטריד את הרכיב שיצרת עם נתונים זמניים כדי לא להגדיל את העומס על בסיס הנתונים שלך ולהאט את מהירות הגישה.

- מנהל מאפיין משותף (Shared Property Manager) - השיטה האחרונה לאחסון נתוני מצב, מנהל המאפיין המשותף, פועלת כמשאב מרכזי או Dispenser שניתן להשתמש בו כדי לשתף ולאחסן נתוני רכיבים. על ידי שימוש בגישה היררכית כלפי מאגר תכונות, מנהל המאפיין מונע סתירות בין הרכיבים והנתונים.

## Caching ו-JIT

עד עתה קראת אודות תוכן אובייקט (object context), מופע (instance) ומצב (state). החלק החשוב הבא במבוך MTS הוא הטמנה (caching). כפי שהזכרתי קודם לכן, MTS מנהל "מאגר" (Pool) של מופעי רכיבים שהגישה אליהם היא על ידי תכני אובייקט בתוך היישומים שלך. המהדורה הנוכחית של MTS מכילה תמיכה יסודית בהטמנת רכיבים, וגרסאות עתידיות יתמכו באופן מלא בהטמנת רכיבים ב-MTS, לצורך הפעלה מהירה מחדש. במהדורת MTS הנוכחית (2.0), הרכיבים יוטמנו כל זמן שיישום מסוים ישתמש בהם. למעשה, הרכיבים נשארים במטמון במשך כשלוש דקות מהרגע שהיישום מסיים את ההתייחסות אליהם. כשרכיב מוטמן, ניתן ליצור במהירות מופעים חדשים שלו, על ידי הפעלת JIT (Just In Time). JIT מאפשרת ל-MTS (וליישומים שלך) לשפר את הביצועים של מופעי רכיב חדשים, בניגוד להתחלה מחדש בכל פעם. הפעלת JIT מוצגת בתרשים 21.3.

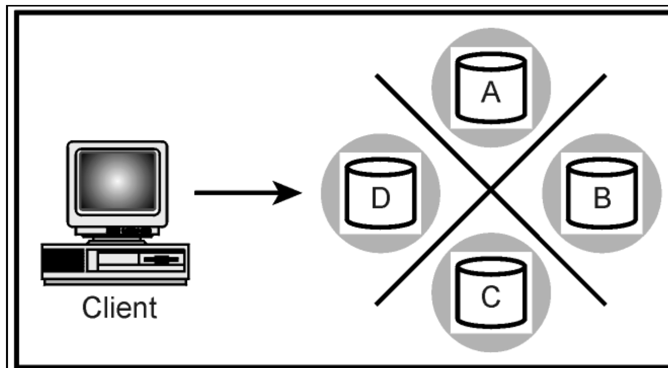



**תרשים 21.3** הפעלת Just-inTime מאפשרת לך "להחיות" רכיבים מחוקים במהירות.

# בידוד תהליך עיבוד

יתרון נוסף של הוספת תוכן עיקרי לרכיבים הוא בידוד תהליך העיבוד. Microsoft מתבססת על העיקרון ש-Windows NT היא מערכת הפעלה חסינה אשר יוצרת בידוד בין תוכניות במטרה לשפר את היציבות. ברמה הנמוכה ביותר, נעשה דבר זה על ידי

בידוד תהליך העיבוד, אשר מאפשר לתוכניות להתייחס למרחב שלהן כאל אי בודד ולהרחיק ממנו זרים. תוכן הרכיב מספק ליישומי האינטרנט שלך את אותה העוצמה על ידי בידוד מופע רכיב ממשנהו. תרשים 21.4 מדגים בידוד תהליך עיבוד בתוך MTS.




**תרשים 21.4** בידוד תהליך העיבוד מוסיף רמה של אמינות ויציבות ליישומים שלך.

# הבנת תנועות

אנו חיים בעולם של תנועות (Transactions). כל מה שאתה עושה יש לו מצב התחלתי ומצב סופי, אשר מסמלים את תחילת וסיום כל פעולה שאתה מבצע. לרוע המזל, לא כל המפתחים שומרים על שטף עבודה בסגנון התחלה-אמצע-סיום כשהם מפתחים יישומי אינטרנט. בעולם המושלם, כל פעולה שהתחילה היתה מסתיימת במלואה. עם זאת, כמו שכל מפתח אינטרנט יודע, איננו חיים (או עובדים) בעולם מושלם.

Microsoft Transaction Server (MTS) פותח במטרה להביא בקרת תנועות (transactional control) לפלטפורמת Windows. בקרת תנועות (Transactional Control) אינה פעולה חדשה לתעשיית המחשוב. שרתי תנועות כגון CICS ו-Tuxedo היו זמינים עבור פלטפורמות אחרות כבר לפני שנים רבות. הנקודה היא הבנת הפעולה של שרת התנועות. תפקיד השרת הוא לעטוף פעולות בודדות ולשייך אותן לפעולה גדולה יותר, עם ציוני דרך מוגדרים מסוג "go/no go".

כשאתה מגיע לבנק שלך כדי להעביר כסף מחשבון החיסכון לחשבון העו"ש, אתה מבצע (באופן מילולי) תנועה (Transaction, באופן מילולי). תנועה זו, למעשה, מורכבת מפעולות שונות המקובצות יחדיו ליחידה אחת, שורה אחת, אם תרצה, בדף החשבון שלך. להלן השלבים שמאחורי פעולה זו:

1. אתה יוצר דרישה להעברה על ידי ציון חשבון המקור (חשבון החיסכון שלך) וחשבון היעד (חשבון העו"ש). אתה מציין גם את הסכום שיש להעביר בין שני החשבונות.
2. ראשית, הבנק בודק את היתרה בחשבון החיסכון שלך כדי להבטיח שאכן יש לך יתרה מספיקה כדי לבצע את ההעברה.

3. לאחר שהבנק וידא שהיתרה בחשבון החיסכון מספיקה, הוא מאשר את החשבון להעברה.

4. לאחר שהבנק וידא שחשבון היעד קיים, הוא מעביר את כמות הכסף המצוינת מחשבון החיסכון לחשבון העו״ש.

5. לאחר מכן, הבנק בודק את היתרה בחשבון העו״ש כדי לוודא שהכסף אכן הועבר אליו מחשבון החיסכון.

6. לאחר קבלת האישור, הבנק מחייב את חשבון החיסכון שלך בסכום שהועבר ומפחית את היתרה בחשבון זה.

7. הפעולה כולה הושלמה, ואתה מקבל אישור מודפס על כך.

כשחילקנו את העברת הכסף לשבעה שלבים, היו לפעולה שבע אבני דרך ייחודיות, או מצבי ״go/no go״. אם כל אחת מאבני הדרך הללו היתה מחזירה שגיאה (כגון יתרה לא מספיקה לצורך ביצוע ההעברה בשלב 2), הפעולה בחשבון היתה מבוטלת ולא היה מועבר כסף. אם היתה בעיה עם ההעברה הממשית של כסף בשלב 4, שלב 5 היה מציין שההעברה לא הצליחה, ולכן לא היה נגרע כסף מחשבון המקור שלך (חשבון החיסכון).

בצורה זו, הפעולה מספקת אבטחת נתונים, ומבטיחה שכל השלבים המתאימים בתהליך הכולל יושלמו בהצלחה לפני שניתן יהיה להכריז על הפעולה כמוצלחת. אם, מכל סיבה שהיא, התהליך נכשל, לא יתבצעו שינויים והפעולה תבוטל.

# הבנת ACID

כדי שהפעולה (במקרה זה העברת כסף מחשבון לחשבון) תצליח, אתה מסתמך על קו מנחה מוסכם (תהליך) שיוצר סדרה מושלמת של פעולות. לרוב, משמעות הדבר היא שכל פעולה, שמהווה חלק מהתהליך, מושלמת בהצלחה (למרות שלא תמיד זהו המקרה). בעולם תהליכי עיבוד התנועות, הביטוי ACID מתייחס לביצוע פעולות מוצלחות. משמעות ACID (Atomic, Consistent, Isolated, Durable) היא:

- Atomic - כדי שהתהליך ייחשב Atomic, כל הפעולות שבו צריכות להסתיים בהצלחה, או שהתהליך לא ייחשב כמושלם.
- Consistent - תהליך עקבי מבטיח כי כל הנתונים שהוא מטפל בהם לא יפריעו לשלמות הכוללת של המערכת (כגון בסיס נתונים).
- Isolated - תוצאות הפעלת כל הפעולות שמרכיבות את התהליך יתנו את אותה התוצאה כמו בהפעלת כל אחת מהפעולות בנפרד, על פי סדר הפעלתן.
- Durable - פעולות שהתוצאות שלהן מאוחסנות על גבי התקנים יציבים (כגון דיסק קשיח או בסיס נתונים) לפני שהתהליך מוגדר כמוצלח.

כשאתה מתכוון להחיל את חוקי ACID על התהליכים שלך, עליך לזכור תמיד שעקרונות מנחים אלה באים להבטיח את שלמות היישום שלך. אם אינך מסכים לעקרונות אלה, אתה מסתכן בתהליכים שעשויים לפגוע בשלמות היישום שלך או בסביבת ההפעלה.

## Commit and Abort

לתנועות (Transactions) יש שתי תוצאות אפשריות: הצלחה או כישלון. במונחי עיבוד תנועות, הצלחה נקראת בשם **חיוב** (Commit). דבר זה מציין כי סך כל הפעולות בתנועות יהיו מחויבות לסביבת ההפעלה, כגון בסיס נתונים. מצד שני, כישלון בביצוע תנועה נקרא **ביטול** (Abort). דבר זה מציין כי כל הפעולות שהתבצעו על ידי התנועה מבוטלות, וסביבת ההפעלה משחזרת את המצב בו היא היתה לפני תחילת ביצוע התנועה הכושלת. MTS משתמש בשני אירועים אלה (Commit ו-Abort) כדי לקבוע את מסלול הפעולה לאחר השלמת התנועה (Transaction).

אירוע Commit נקבע על ידי שיטת SetComplete ואילו אירוע Abort נקבע על ידי SetAbort. מבחינת התכנות, אתה יכול למנוע מ-MTS ו-DTC לאפשר חיוב או ביטול תנועה. אתה עשוי להזדקק לאפשרות זו כשהיישום שלך מטפל בנתונים או מבצע פעולות משלו. שיטת DisableCommit מונעת מ-MTS, DTC ו-RM לחייב תנועה (Transaction) עד לקריאה לשיטת EnableCommit.

## תנועות מבוזרות

חשוב להכיר בכך ש-MTS אינו מוגבל לניהול תנועות (Transactions) בשרת המקומי שלך. MTS מספק תמיכה מלאה בתנועות שמתבצעות על פני כמה מערכות, גם על גבי רשות, כל זמן שקיימים שירותי תנועה (כגון Microsoft Message Queuing Services, המתוארים בפרק הבא). לרוב, התנועות שלך יבצעו פעולה יחד עם שרת חיצוני, כגון בסיס הנתונים המפעיל Microsoft SQL או Oracle. כדי לטפל בתנועות מבוזרות, MTS מסתמך על שירות נפרד שנקרא DTC (Distributed Transaction Coordinator). DTC מותקן כחלק מ-Microsoft SQL Server ו-MTS.

DTC (כפי ששמו מציין) שולט בתנועות (Transactions) בין מערכות על ידי מנהלי משאבים (Resource Managers - RM). מערכות בסיסי נתונים, כגון Microsoft SQL Server, מטמיעות RM אשר פועל יחד עם DTC לצורך ניהול תנועות. התהליך פשוט למדי ברמה הגבוהה:

1. היישום שלך (הלקוח) מתקשר עם אובייקט התוכן שלו בתוך MTS ומאתחל תנועה.
2. אובייקט התוכן מתקשר עם הרכיב שלך ומתחיל את תהליך התנועה.

3. לאחר מכן, MTS קורא ל-DTC כדי לגשר את התנועה על גבי מערכת אחרת, כגון שרת בסיס הנתונים.

4. DTC מתקשר עם RM של המערכת החיצונית (בסיס הנתונים) בנוגע לתנועה.

5. לאחר ש-MTS קבע שהתנועה הצליחה, הוא מעביר אירוע חיוב (Commit) ל-DTC ול-RM. לחילופין, אם התנועה נכשלה, מועבר אירוע ביטול (Abort) ל-DTC ול-RM.

6. על פי האירוע שהועבר ל-RM, הוא מבצע את השינויים הדרושים (במצב חיוב) או מבטל את הפעולות וחוזר למצב שהיה לפני תחילת התנועה (במצב ביטול).

השימוש ב-DTC מאפשר ל-MTS לגשר על פני כמה מערכות כל זמן שהשירותים החיצוניים הללו מציעים RM תואם DTC.

## COM+ ועתיד MTS

בדיוק כפי ש-COM התפתח משירות האב שלו (OLE) על פני השנים, הוא ממשיך להתפתח לפלטפורמה רבת עוצמה: COM+. Microsoft רואה ב-COM+ את השלב המהפכני הבא של מודלי אובייקטי הרכיבים, כולל שילוב ושיפור תשתית ה-MTS. COM+ היא תכונה פנימית של Windows 2000 והיא פועלת כבסיס לתכונות מתקדמות של מערכות הפעלה רבות. תכונות COM+ כוללות:

- איזון עומס דינמי שמיועד לשפר את יכולת ההרחבה של הרכיבים שלך ויישומים מבוססי רכיבים.
- בסיס נתונים הנמצא בזיכרון שמשפר את ביצועי הרכיבים.
- יכולת לכלול אירועי הפצה/פרסום (Publish) ורישום (Subscribe) וכן תורי רכיבים, ובכך מתאפשר שילוב קל יותר עם כמה רכיבים.
- מודל תכונות מפושט שמשלב MTS ו-COM.

**הערה:**

למידע נוסף אודות COM והשיפורים שהוא מציע כחלק מ-Windows 2000, פנה לאתר COM של Microsoft בכתובת http://www.microsoft.com/com.

# MTS ורכיבים

כדי להשתמש ברכיבים שלך יחד עם MTS, אינך חייב לבצע שינויים בקוד שלך. עם זאת, כדי להכיר מעשית ביתרונות ה-MTS, על הרכיבים שלך לאמץ "שיטות מיטביות" לעיצוב רכיבים. בסעיף זה, תפתח רכיב בסיסי ותציב אותו תחת בקרת MTS. סעיף זה דן גם בעקרונות ניהול הרכיבים שלך תחת MTS. הסעיף הבא פותח את הדלת לגישה לרכיבים שמודעים ל-MTS בתוך דפי ה-ASP שלך.

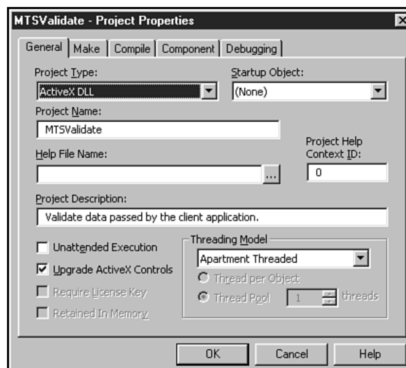
## יצירת רכיב MTS

בדוגמה זו, תיצור רכיב MTS פשוט תוך שימוש ב-Visual Basic. תתמקד בפרטי הפיכת הרכיבים שלך למודעים ל-MTS. אם אתה זקוק למבוא ליצירת רכיבים, עליך לעבור לפרק 18. ההוראות בסעיף זה יספקו לך גם את המידע הדרוש לכתיבת הרכיב בשפות תכנות אחרות, כגון ++C.

התחל בתכנון הרכיב שלך וקבע את תכונותיו. לצורך דוגמה זו, הרכיב יקרא פרמטרים המועברים אליו על ידי היישום הקורא (כגון דף ASP), יבצע פעולה לוגית פנימית, ויחזיר ערך. בהתבסס על הפרמטרים המועברים אליו, הרכיב יקבע אם התנועה הצליחה או נכשלה.

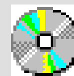
כדי לפתח את הרכיב, הפעל את Visual Basic (גירסה 5.0 ומעלה) ובצע את השלבים הבאים:

1. צור פרויקט ActiveX DLL חדש מתוך תיבת הדו-שיח New Project.
2. לחץ לחיצה ימנית על מודול המחלקה Class1 ב-Project Explorer ובחר באפשרות Properties מתוך תפריט הקיצור.
3. בתכונת Name, הקלד את השם ValidateData עבור שם המחלקה.
4. לחץ לחיצה ימנית על הפרויקט שלך ב-Project Explorer ובחר באפשרות Project Properties מתוך תפריט הקיצור כדי לפתוח את תיבת הדו-שיח Project Properties המוצגת בתרשים 21.5.
5. בתיבת הטקסט Project Name, קרא לפרויקט בשם MTSValidate. כעת, זהו שם הרכיב שלך.
6. בתיבת הטקסט Project Description, הקלד תיאור קצר של הרכיב: Validate data passed by the client application.




**תרשים 21.5** תיבת הדו-שיח Project Properties משמשת להגדרת מאפיינים לפרויקט Visual Basic כולו.

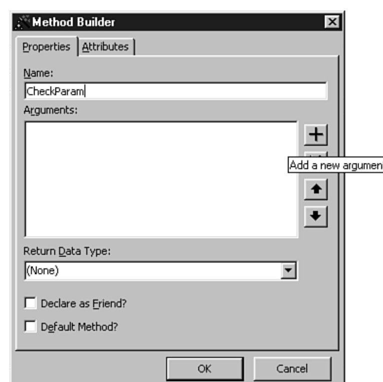
## הערה:



קובץ פרויקט Visual Basic mtsvalidate.vbp נמצא בתקליטור המצורף. הוא כולל התייחסויות לרשימות הקוד המלאות של הרכיב אותו אתה מפתח.

השלב הבא הוא להוסיף את קוד ה-COM. לשם כך, עלינו להשתמש ב-VB Class Builder. זוהי תוכנית שירות הכלולה ב-Visual Basic. ניתן להגיע אל תוכנית שירות זו מתוך תפריט Add-In. עם זאת, ייתכן ויהיה עליך להתקין אותה אם לא עשית זאת בזמן התקנת Visual Basic. לאחר שפתחת את Class Builder מתוך תפריט Add-Ins, בצע את השלבים הבאים:

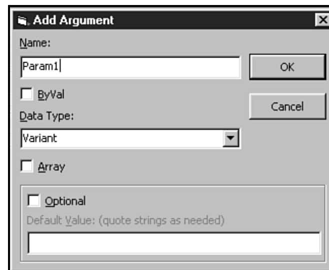
**1.** לחץ על לחצן Add New Method to Current Class בסרגל הכלים כדי לפתוח את תיבת הדו-שיח Method Builder המוצגת התרשים 21.6.




**תרשים 21.6** תיבת הדו-שיח Method Builder מפשטת את תהליך הוספת השיטות לרכיבים שלך.

**2.** בתיבת הטקסט Name, תן לשיטה את הערך CheckParam.

**3.** זה הזמן להוסיף ארגומנטים חדשים לשיטה זו. לחץ על לחצן Add a New Argument (לחצן +) כדי לפתוח את תיבת הדו-שיח Add Argument המוצגת בתרשים 21.7.




**תרשים 21.7** בדוגמה שלך תשתמש בסוגי נתונים פשוטים מסוג Variant עבור הארגומנטים שלך.

**4.** בתיבת הטקסט Name, תן לארגומנט את השם Param1.

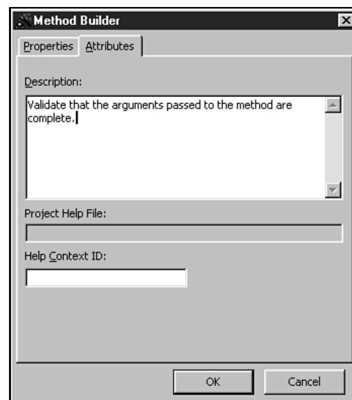
**5.** לחץ על לחצן OK כדי להוסיף את הארגומנט החדש.

**6.** חזור על שלבים 3-5 וצור ארגומנט שני בשם Param2.

**7.** בחר בערך String בתיבת הרשימה Return Data Type.

**8.** סמן את תיבת הסימון Default Method כדי לציין שיש להשתמש בשיטה זו אם לא צוינה שיטה בקריאה לרכיב זה.

**9.** בחר בכרטיסיה Attributes בראש החלון כדי לעבור לדף השני של תיבת הדו-שיח, המוצג בתרשים 21.8.




**תרשים 21.8** הכרטיסיה Attributes מאפשרת לך לתאר את השיטה.

**10.** בתיבת הטקסט Description, הקלד את תיאור השיטה הבא:

Validate that the arguments passed to the method are complete.

**11.** לחץ OK כדי לבצע את השינויים שהגדרת לשיטה זו. השיטה החדשה תופיע ב-Class Builder.

**12.** מתוך סרגל התפריטים, בחר באפשרות Update Project מתוך תפריט File כדי להוסיף את השינויים שלך לקובץ המחלקה ValidateData.

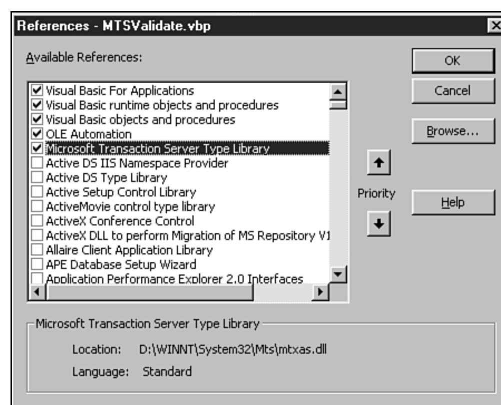
**13.** צא מתוכנית השירות Class Builder על ידי בחירת האפשרות Exit מתוך תפריט File בסרגל התפריטים.

מודול המחלקה שלך אמור להכיל כעת את שתי שורות הקוד הבאות:

```
Public Function CheckParam(Param1 As Variant, Param2 As Variant)
 ↳As String
End Function
```

לפני שתוכל להוסיף את הקוד עבור שיטת CheckParam, עליך להוסיף התייחסות לספריית הסוגים של Microsoft Transaction Server. דבר זה יאפשר ל-Visual Basic ולרכיבים שלנו להשתמש בפונקציונליות ה-MTS. לשם כך, בצע את השלבים הבאים:

**.1** בחר באפשרות References מתוך תפריט Project כדי לפתוח את תיבת הדו-שיח References, כפי שמוצג בתרשים 21.9.




**תרשים 21.9** כדי להשתמש ב-MTS ברכיבים שלך, עליך להתייחס לספריית הסוגים של MTS.

**.2** אתר את ספריית הסוגים של MTS ברשימה:
Microsoft Transaction Server Type Library.

**.3** סמן את תיבת הסימון של ספריית הסוגים של MTS כדי לציין שייעשה בה שימוש בפרויקט.

**.4** לחץ על לחצן OK כדי להוסיף את ההתייחסות הזו ולחזור לעורך Visual Basic.

כעת, הוסף את הקוד עבור הפונקציה. החלף את קוד הצהרת הפונקציה בתדפיס 21.1.

**תדפיס 21.1** קוד השיטה MTSValidate.CheckParam.

```
Public Function CheckParam(Param1 As Variant, Param2 As Variant) As String
    Dim objContect As ObjectContext
    Set objContext = GetObjectContext()
    If Param1 <> Param2 Then
        objContext.SetComplete
    Else
        objContext.SetAbort
    End If
End Function
```

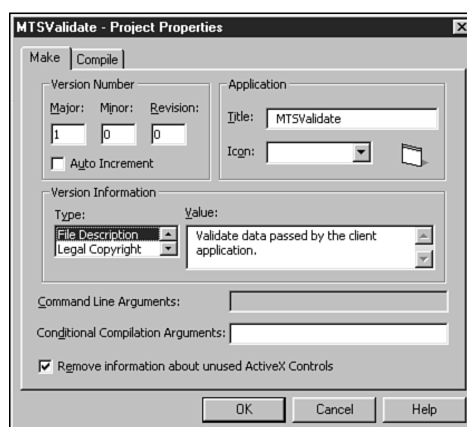
רכיב פשוט זה יקרא שני פרמטרים המועברים על ידי הארגומנטים Param1 ו-Param2. אם שני הארגומנטים אינם תואמים (כלומר, כל אחד מהם ייחוסי), הרכיב קורא לשיטה SetComplete בשורה 5, ומציין בכך שהתנועה הצליחה. עם זאת, אם שני הארגומנטים תואמים, מתבצעת קריאה לשיטה SetAbort (שורה 7) המציינת שהתנועה נכשלה.

**הערה:**

זוהי, כמובן, דוגמה פשוטה מאוד לדרך העבודה של MTS בתוך רכיב. ללא ספק, תוכל לחשוב על יישומים טובים יותר של MTS בעבודתך.

כדי להשלים את השלב הזה, עליך להדר את הרכיב שלך בצורת ActiveX DLL שיהיה מוכן לשימוש על ידי MTS. לשם כך, בצע את השלבים הבאים:

1. בחר באפשרות Make MTSValidate.dll מתפריט File שבסרגל התפריטים כדי לפתוח את תיבת הדו-שיח Make Project.
2. לחץ על לחצן Options כדי לפתוח את תיבת הדו-שיח Project Properties, המוצגת בתרשים 21.10.




**תרשים 21.10** תיבת הדו-שיח Project Properties, בזמן ההידור היא סדרת משנה של תיבת הדו-שיח Project Properties בזמן העיצוב.

3. בתיבת הרשימה Type, אתר את הפריט File Description ובחר בו.
4. בתיבת הטקסט Value, הקלד תיאור עבור הרכיב הזה:
Validate data passed by the client application.
בתיאור זה תשתמש כדי לזהות את הרכיב שלך בהתייחסויות שבתוך הפרויקט.
5. לחץ OK כדי לחזור אל תיבת הדו-שיח Make Project.
6. לחץ OK כדי להדר את הרכיב החדש שיצרת.

**הערה:**

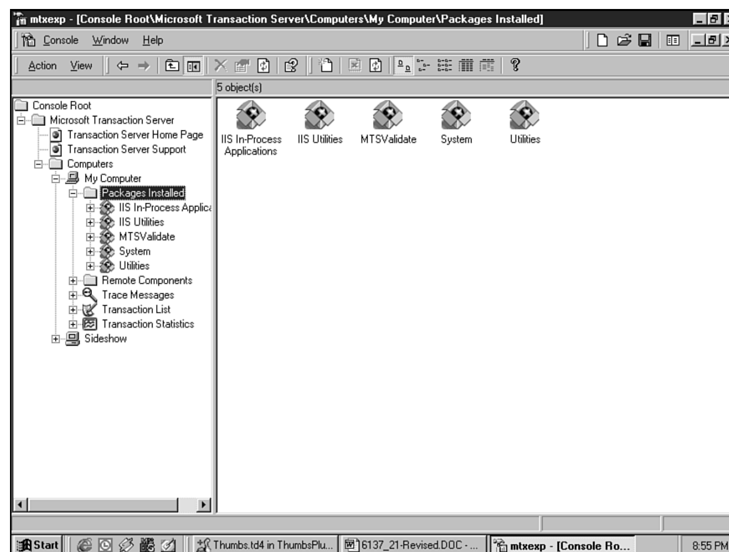
אם אתה מתכנן להדר את הרכיב שלך יותר מפעם אחת, עליך לאפשר את האפשרות Binary Compatibility בתיבת הדו-שיח Project Properties. האפשרות Binary Compatibility תבטיח כי ActiveX DLL יירשם ברישום המערכת של Windows (Registry) עם אותם הנתונים, בניגוד ליצירת רישום חדש בכל פעם. אם האפשרות אינה מאופשרת, השינויים שתבצע ברכיב לא יוכרו על ידי יישומים שמתייחסים למופע הישן שלו.

# אריזת רכיבים בעזרת MTS Explorer

לפני שתוכל להשתמש ברכיב ביישום שלך, עליך לגרום לכך ש-MTS יהיה מודע לו. כשאתה מיידע את MTS בקיום הרכיב החדש, אתה **אורז** את הרכיב ומכין אותו לשימוש. למעשה, אריזות MTS עשויות להכיל כמה רכיבים ולהתנהג כקבוצה לוגית. קרוב לוודאי שתשתמש באריזות כדי לאחסן את כל הרכיבים הקשורים ביישום מסוים. האריזה מתבצעת ב-Microsoft Transaction Server Explorer (MTS Explorer), המבוסס על MMC (Microsoft Management Console). MTS Explorer מותקן בזמן התקנת MTS במערכת.

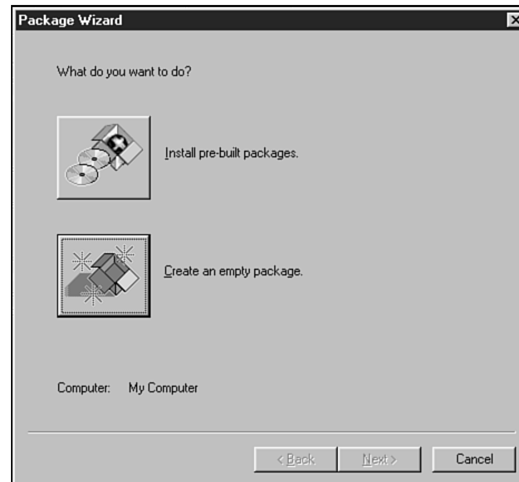
כדי לארוז את הרכיב שלך, בצע את השלבים הבאים:

1. פתח את MTS Explorer מתוך תפריט Start.
2. הרחב את רשימת התיקיה ב-Console Root כך שתציג את המחשב שלך (My Computer) ואת תיקיית PackagesInstalled, כפי שמוצג בתרשים 21.11.



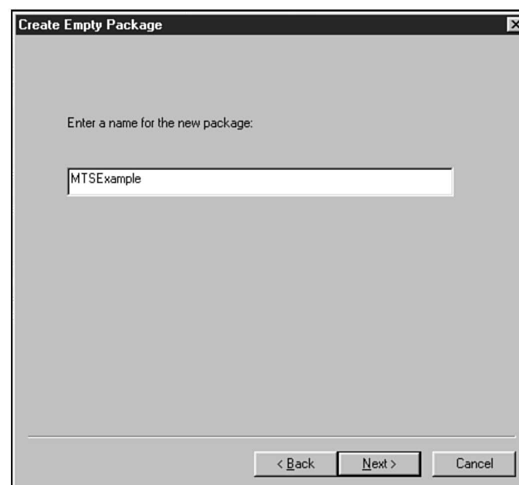

**תרשים 21.11** אריזות הן קיבוץ לוגי של רכיבים

**3.** לחץ לחיצה ימנית על תיקיית PackagesInstalled ובחר באפשרות New, Package מתפריט התוכן כדי לפתוח את המסך הראשון של ה-Package Wizard, המוצג בתרשים 21.12.



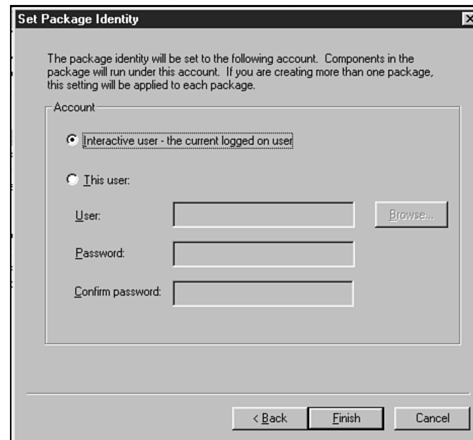

**תרשים 21.12** הוספת אריזות חדשות ל-MTS באמצעות Package Wizard.

**4.** אתה בונה אריזה חדשה, ולכן עליך ללחוץ על לחצן Create an Empty Package כדי לעבור לדף השני של האשף, המוצג בתרשים 21.13.




**תרשים 21.13** כל שלב באשף מבקש נתונים הקשורים באריזה שלך.

**5.** בתיבת הדו-שיח Create Empty Package, הקלד שם עבור האריזה החדשה: MTSExample, ולחץ על לחצן Next כדי לעבור למסך הבא המוצג בתרשים 21.14.




**תרשים 21.14** כשהרכיבים שלך פועלים מתוך דף ASP, האפשרות העליונה תכריח את הרכיבים לפעול כמשתמש עם שם מחשב _IUSR.

**6.** הדף השלישי של האשף, Set Package Identity, משמש להגדרת חשבון המשתמש במערכת דרכו יופעלו הרכיבים שבאריזה. בדוגמה זו, בחר ב:

Interactive User - The Current Logged on User.

**7.** לחץ על לחצן Finish כדי להוסיף את האריזה החדשה. אריזת MTSExample תופיע תחת תיקיית PackagesInstalled במסוף.

לאחר יצירת האריזה, עליך להוסיף את הרכיב MTSValidate עצמו ל-MTS. לשם כך, בצע את השלבים הבאים:

**1.** הרחב את תיקיית האריזה MTSExample למסוף כדי להציג את תיקיית המשנה Components.

**2.** לחץ לחיצה ימנית על תיקיית Components ובחר באפשרות New, Components מתוך תפריט התוכן כדי לפתוח את תיבת הדו-שיח Component Wizard המוצגת בתרשים 21.15.

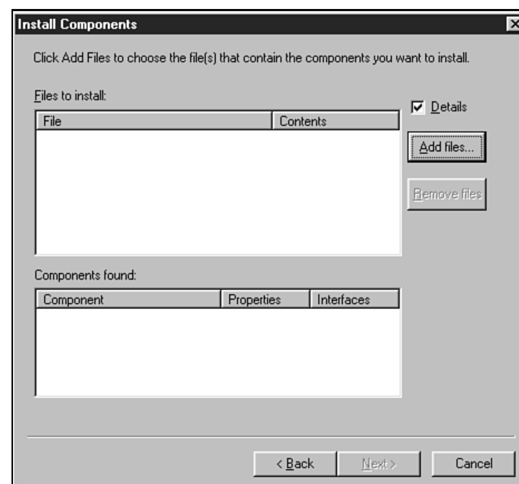
**תרשים 21.15** Component Wizard מזכיר מאוד את Package Wizard.

**.3** דף זה מציע שתי אפשרויות: Install New Component(s) ו-Import Component(s) That Are Already Registered. אנו מניחים שזהו רכיב חדש, ולכן עליך ללחוץ על לחצן Install New Component(s).

## הערה:

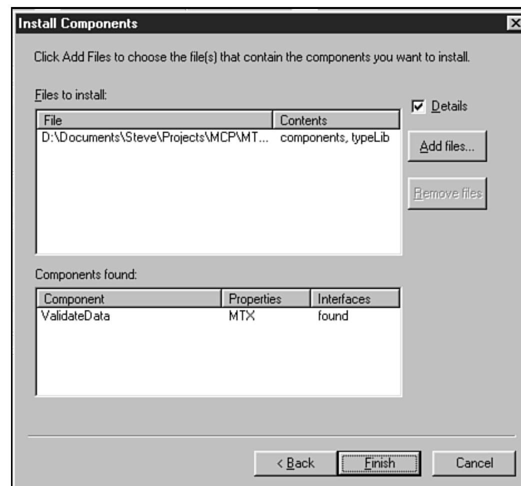
שתי האפשרויות המוצעות משיגות את אותה המטרה תחת נסיבות שונות. לחצן Install New Component(s) מיועד להתקנת רכיב שלא נרשם לפני כן באותה המערכת בה MTS פועל. האפשרות השנייה, Import Component(s) That Are Already Registered, מיועדת להוספת רכיב שכבר נרשם במערכת ההפעלה כשרת COM. אם אתה אורז את הרכיב באותה המערכת בה פיתחת אותו, תוכל לבחור באפשרות השנייה. עם זאת, לרוב תתקין רכיב בשרת הפועל על מחשב שונה מזה בו פיתחת את הרכיב. במצב מעין זה עליך לבחור באפשרות הראשונה. בדוגמה זו, הנח כי מדובר בשרת "נקי".

**.4** כעת יופיע הדף השני של האשף, Install Components (כפי שמוצג בתרשים 21.16). לחץ על לחצן Add Files כדי לפתוח את תיבת הדו-שיח Select Files to Install.




**תרשים 21.16** תוכל להתקין כמה רכיבים בבת אחת.

**.5** אתר את העותק שלך של קובץ MTSValidate.dll ולחץ על לחצן Open. הרכיב שלך יתווסף לרשימה, כפי שמוצג בתרשים 21.17.




**תרשים 21.17** פרטי הרכיבים אותם בחרת יוצגו בלוח התחתון.

**6.** לחץ על לחצן Finish כדי להוסיף את הרכיב.

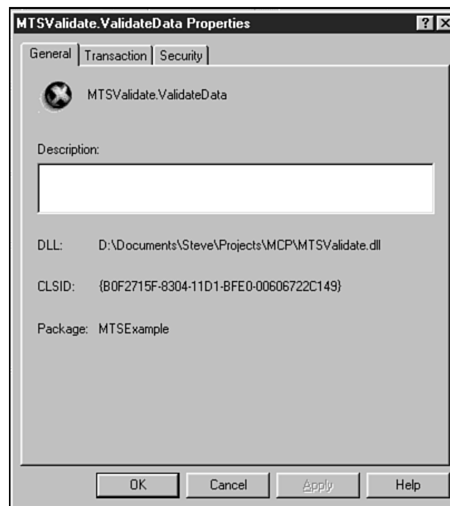
לאחר שהרכיב הותקן כחלק מהאריזה, הוא זמין לשימוש על ידי MTS והיישומים שלך. עם זאת, נשאר עוד שלב אחד לביצוע. כדי שתנועה תוכל להפעיל את MTS יש להורות ל-MTS כיצד להתייחס לרכיב שלך ולתמיכת התנועות (Transactions) שבו. תמיכת תנועות מגיעה עבור הרכיבים בארבע צורות:

- Requires a transaction - הרכיב חייב שתהיה לו פונקציה ליצירת תנועה כדי לתפקד כהלכה. אם תתבצע קריאה לרכיב בתוך תנועה, הוא ישתמש בתנועת ההורה. אם לא קיימת תנועה, הרכיב ייצור תנועה חדשה.
- Requires a new transaction - ללא קשר אם הקריאה לרכיב התבצעה מתוך תנועה קיימת או לא, הרכיב ייצור לעצמו תנועה חדשה.
- Supports transactions - הרכיב ישתמש בתנועה חיצונית אם קיימת כזו, אך לא ייצור תנועה חדשה אם לא קיימת תנועה חיצונית.
- Does not support transactions - הרכיב אינו תומך בתנועות ולא ייכלל בהן.

כדי להגדיר כיצד MTS ינהג ברכיב שלך, בצע את השלבים הבאים:

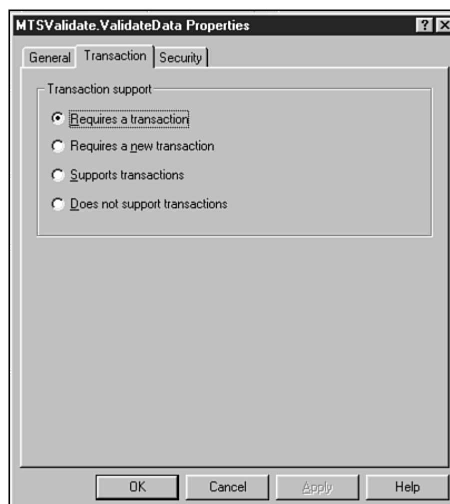
**1.** לחץ לחיצה ימנית על MTSValidate.ValidateData תחת תיקיית Components במסוף.

**2.** בחר באפשרות Properties מתוך תפריט התוכן כדי לפתוח את תיבת הדו-שיח MTSValidate.ValidateData Properties, כפי שמוצג בתרשים 21.18.




**תרשים 21.18** תוכל להקליד כאן תיאור של הרכיב שלך ב-MTS.

**3.** בחר בכרטיסיה Transaction בראש תיבת הדו-שיח, כפי שמוצג בתרשים 21.19.




**תרשים 21.19** תמיכת התנועות יכולה להיות באחת מארבע צורות.

**4.** בחר בלחצן האפשרויות Transaction כדי להורות ל-MTS שהרכיב שלך דורש תנועה כדי לפעול.

**5.** לחץ OK כדי להחיל את השינויים שביצעת.

כעת הרכיב שלך זמין ל-MTS ולישומים שלך!

# שימוש ב-MTS להגדלת יכולת ההרחבה והביצועים של ASP

הוספת MTS לערבוביית יישומי האינטרנט והרכיבים שלך מספקת גמישות רבה, יכולת הרחבה ועוצמה. על ידי שימוש ב-MTS, תוכל להוסיף סבילות כישלון ליישומי האינטרנט שלך על ידי הבטחה שהרכיבים זמינים ומנוהלים בצורה יעילה, הפעולות יקרו כפי שמצופה (הבטחת שלמות), ויישום האינטרנט שלך יוכל לגדול כדי לטפל בכמה שרתים, משתמשים רבים נוספים וסביבה מבוזרת.

**הערה:**

אני אדון ברמה אחרת של יכולת הרחבה ואמינות, אותם אתה יכול להחיל על יישומי האינטרנט שלך, בפרק 22.

בסעיף זה, תגלה שתי שיטות שונות לשילוב MTS ביישומי האינטרנט שלך:

- על ידי יצירת Transactional Active Server Pages
- על ידי קריאה לרכיבים מודעי MTS ב-ASP

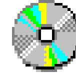
ניתן לשלב כל אחת משיטות אלו ביישומי האינטרנט שלך כדי ליצור פלטפורמות יישומים רבות עוצמה.

## יצירת Transactional Active Server Pages

למרות ש-MTS מתאים במיוחד לשימוש ברכיבים מותאמים אישית, מפתחי ASP ויישומי אינטרנט יכולים לנצל את הגמישות והעוצמה של MTS בקלות רבה. קל מאוד לכלול MTS בדפי האינטרנט שלך. עם זאת, הדבר דורש הבנה של גישת התנועות וטיפול בתכנון לוגיקת התסריטים שלך. ארבעת השלבים ביצירת תנועות (Transactions) בתוך דפי ה-ASP שלך הם:

1. הצהר על הדף שלך כדף בר-תנועה (Transactional), כדי שהשליטה בו תיעשה על ידי MTS.
2. פתח את התסריטים שיהיו חלק מהתנועה.
3. הגדר מהי תנועה מוצלחת או תנועה כושלת ואת נקודות "אבני הדרך".
4. צור אירועי תנועה המטפלים בתנועה מוצלחת ותנועה כושלת.

# יישום לדוגמה



בפרק זה, תיצור יישום פשוט שיקבל את הקלט שלו מהמשתמש בעזרת טופס ויאחסן את הנתונים בבסיס הנתונים. לצורך ביצוע דוגמה זו, תשתמש בדוגמת בסיס הנתונים Pubs הכלולה ב-Microsoft SQL Server גירסה 6.5 ו-7.0. אם אינך משתמש ב-Microsoft SQL Server או שבסיס הנתונים Pubs לא נמצא ברשותך, תוכל ליצור את הטבלה היחידה בה תשתמש יחד עם תסריט SQL המוצג בתדפיס 21.2. תסריט SQL זה נמצא גם בתקליטור הנלווה תחת השם storesdb.sql.

**תדפיס 21.2** תסריט Stores Database SQL

```
Create Table dbo.stores
(
   stor_id char(4) Not Null,
   stor_name varchar(40) Null,
   stor_address varchar(40) Null,
   city varchar(20) Null,
   state char(2) Null,
   zip char(5) Null
)
Go
Alter Table dbo.stores Add Constraint
    UPK_storeid Primary Key Clustered
(
   stor_id
)
Go
```

תסריט זה יוצר את בסיס הנתונים stores על ידי שימוש בשש עמודות (stor_id, stor_name, stor_address, city, state, zip) וקובע את עמודת stor_id כמפתח העיקרי של הטבלה.

**הערה:**

אם אתה משתמש ב-Microsoft SQL Server, תוכל להשתמש ב-Query Analyzer כדי לבצע את התסריט. אם אתה משתמש בשרת בסיס נתונים של יצרן אחר, פנה לתיעוד בסיס הנתונים לקבלת הנחיות לביצוע תסריטי SQL.

צור קובץ DSN מתוך לוח הבקרה ODBC של בסיס הנתונים Pubs (או בסיס הנתונים שמאחסן את הטבלה החדשה אותה יצרת). בנוסף, לפני העיבוד, ודא שלשם המשתמש של שרת האינטרנט שלך (IUSR_machinename) יש גישת בחירה (קריאה) והכנסה (כתיבה) לטבלה.

# הצהרה על דף כ-Transactional

קל מאוד לשלב את MTS בדפים שלך. כשאתה עובד עם MTS, אתה מצהיר כי הדף שלך הוא מסוג transactional וכי כל התסריטים פועלים תחת אותו הקשר תנועה. MTS מנהל את פרטי יצירת התנועה (Transaction) והוא מודע להצלחת הפעולה או כישלונה. בלשון MTS, תנועה מוצלחת נקראת חיוב (Commit), ואילו תנועה כושלת נקראת ביטול (Abort). כדי להצהיר על הדף שלך כעל transactional, יש להוסיף את ההנחיה @TRANSACTION כשורה הראשונה שלו. הנחיית @TRANSACTION מופיעה בצורה הבאה:

```
<% @TRANSACTION = value %>
```

הערך value מתייחס לדרך בה הדף שלך פועל יחד עם MTS. הערכים המותרים מתוארים בטבלה 21.1.

**טבלה 21.1** ערכי @TRANSACTION

| ערך @TRANSACTION | תיאור |
|---|---|
| Requires_New | ערך זה יגרום להתחלת תנועה חדשה עבור הדף |
| Required | דף זה משתמש בתנועה קיימת, אם ישנה כזו. אם אין, הוא יתחיל תנועה חדשה |
| Supported | דף זה אינו יוצר תנועה חדשה, אך הוא יתמוך בתנועות שנוצרו בדף אחר |
| Not_Supported | דף זה אינו תומך בתנועות |

**הערה:**

דפי ה-ASP שלך רשאים להכיל רק שורת הנחיית @ אחת. במידת הצורך, אתה יכול לשלב כמה הנחיות בשורה אחת. לדוגמה:

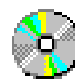
```
<% @TRANSACTION = Required LANGUAGE = VBSCRIPT %>
```

**טיפ:**

רק הדפים שדורשים עיבוד תנועות חייבים לכלול הנחיית @TRANSACTION. שאר חלקי יישום האינטרנט אינם צריכים להצהיר על תמיכת עיבוד תנועות. למעשה, כדי להימנע מבעיות ביצועים, לא כדאי לכלול בהם הנחיה זו.

# יצירת Transaction Scripts

מאחורי הקלעים, דפי ה-ASP שלך יכילו את אותם התסריטים שהיו להם מאז ומעולם. MTS מיועד להרחבת ושיפור התסריטים שלך, אך אין משמעות הדבר שיהיה עליך לשנות את הקוד שיצרת. אתה יכול לקחת כמעט כל דף ASP קיים ולהוסיף לו תמיכה בתנועות (Transactions) מבלי לשנות בהרבה את הקוד שלך. תסריטי התנועות הם הלוגיקה שמאחורי הדף שלך והקוד שמופעל במידת הצורך.



עבור דוגמה זו, תיצור תסריט פשוט שקובע אם יש לעדכן בסיס נתונים במידה שהוא מסופק בטופס. התסריטים שלך ייקחו נתונים מטופס ויאחסנו אותם בבסיס נתונים במידה והתנועה (Transaction) הצליחה (תקבל פרטים נוספים על כך בסעיף מאוחר יותר). כדי להתחיל, עליך ליצור דף נוסף עבור הדוגמה שלך, שנקרא form.htm (דף זה נמצא בתקליטור הנלווה). זהו הדף דרכו יתקבלו הנתונים תוך שימוש בטופס HTML פשוט. כדי ליצור את הקובץ form.htm, בצע את השלבים הבאים:

**1.** צור קובץ חדש בשם form.htm בו תשתמש עבור טופס ה-HTML.

**2.** בעורך שלך, הכנס את תדפיס 21.3 לקובץ.

**תדפיס 21.3** דף form.htm

```
<HTML>
<HEAD>
<TITLE>Transaction Form</TITLE>
</HEAD>
<BODY>
<FONT FACE="Arial">
<H1>Transaction Example: Step 1</H1>
<HR SIZE=1 NOSHADE>
<P ALIGN="LEFT">
     This form will be used to insert data into a database if the
     transaction is successful. If the transaction fails, the user
     will receive a warning message that the action was not successful.
</P>
<HR SIZE=1 NOSHADE>
<FORM action="transaction.asp" method=POST id=frmData name=frmData>
<TABLE BORDER="0" ALIGN="CENTER">
<TR>
<TD>
     Store Name
</TD>
<TD>
<INPUT type="text" id=txtName name=txtName size=20 maxlength=20>
</TD>
</TR>
<TR>
<TD>
```

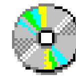
```
        Telephone
</TD>
<TD>
        <INPUT type="text" id=txtTelephone name=txtTelephone size=12
        ↳ maxlength=12>
</TD>
</TR>
<TR>
<TD>
        Street Address
</TD>
<TD>
        <INPUT type="text" id=txtAddress name=txtAddress size=40
        ↳ maxlength=40>
</TD>
</TR>
<TR>
<TD>
        City
</TD>
<TD>
        <INPUT type="text" id=txtCity name=txtCity size=20 maxlength=20>
</TD>
</TR>
<TR>
<TD>
        State
</TD>
<TD>
        <INPUT type="text" id=txtState name=txtState size=2 maxlength=2>
</TD>
</TR>
<TR>
<TD>
        Zip Code
</TD>
<TD>
        <INPUT type="text" id=txtZipCode name=txtZipCode size=5 maxlength=5>
</TD>
<TR>
<TD COLSPAN=2 ALIGN="CENTER">
        <INPUT type="submit" value="Submit" id=btnSubmit name=btnSubmit>
</TD>
</TR>
</TABLE>
</FORM>
<HR SIZE=1 NOSHADE>
<P ALIGN="LEFT">
```

```
73          This page pushes information to the <U>transaction.asp</U>
            ↳ page for processing.
74    </P>
75    </FONT>
76    </BODY>
77    </HTML>
```

**3.** שמור את השינויים שביצעת בקובץ form.htm.

**4.** שמור את השינויים שביצעת בקובץ transaction.asp.

תדפיס 21.3 פשוט מאוד ומובן מעצמו. שורות 15-70 מגדירות את הטופס שיזין נתונים לדף, transaction.asp, כפי שצוין על ידי הפעולה שבשורה 15).



השלב הבא שעליך לבצע הוא ליצור תסריט תנועה משלך בדף transaction.asp המוצג בתדפיס 21.4. קובץ זה נמצא גם על גבי התקליטור הנלווה.

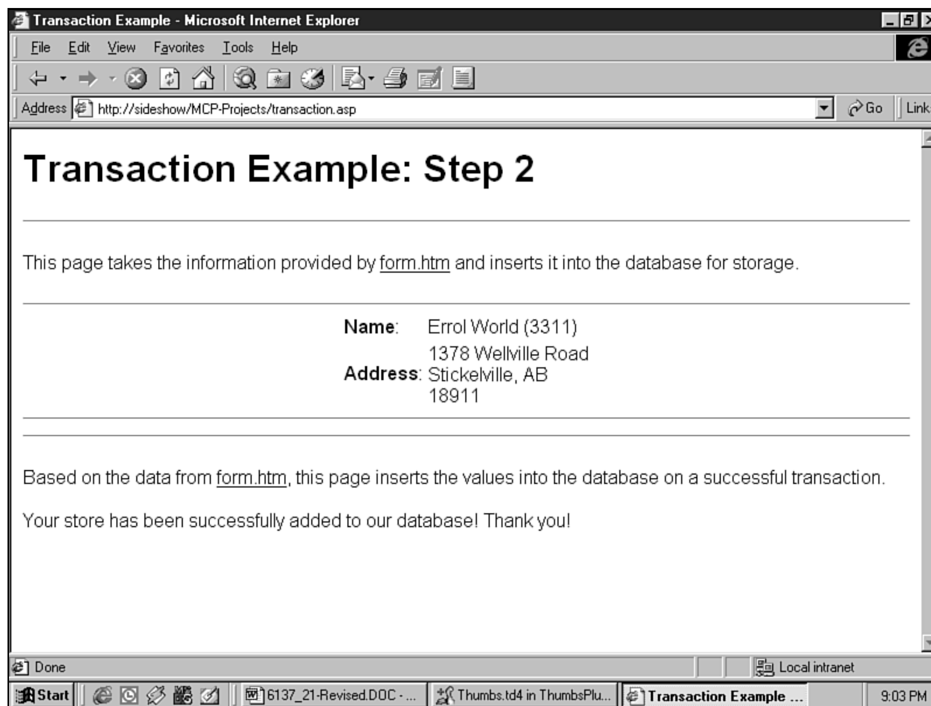
**תדפיס 21.4** תסריט התנועה של קובץ transaction.asp

```
<%@TRANSACTION=Required Language=VBScript %>
<HTML>
<HEAD>
<TITLE>Transaction Example</TITLE>
</HEAD>
<BODY>
<FONT FACE="Arial">
<H1>Transaction Example: Step 2</H1>
<HR SIZE=1 NOSHADE>
<P ALIGN="LEFT">
This page takes the information provided by <U>form.htm</U> and
inserts it into the database for storage.
</P>
<HR SIZE=1 NOSHADE>
<%
Dim strStoreID, strName, strAddress, strCity, strState, strZipCode
Dim objDBConn, strSQLText
strStoreID = Request.Form("txtTelephone")
strName = Request.Form("txtName")
strAddress = Request.Form("txtAddress")
strCity = Request.Form("txtCity")
strState = Request.Form("txtState")
strZipCode = Request.Form("txtZipCode")
strStoreID = right(strStoreID, 4)
%>
<TABLE ALIGN="CENTER">
<TR>
      <TD>
      <B>Name</B>:
      </TD>
      <TD>
```

```
32          <%= strName %> (<%= strStoreID %>)
33          </TD>
34    </TR>
35    <TR>
36          <TD>
37          <B>Address</B>:
38          </TD>
39          <TD>
40    <%= strAddress %><BR><%= strCity %>, <%= strState %><BR>
41    <%= strZipCode %>
42    </TD>
43    </TR>
44    </TABLE>
45    <HR SIZE=1 NOSHADE>
46    <%
47    strStoreID = "'" & strStoreID & "',"
48    strName = "'" & strName & "',"
49    strAddress = "'" & strAddress & "',"
50    strCity = "'" & strCity & "',"
51    strState = "'" & strState & "',"
52    strZipCode = "'" & strZipCode & "'"
53    strSQLText = "INSERT INTO stores (stor_id, stor_name, stor_address," & ␦
54    "city, state, zip) VALUES (" & strStoreID & strName & strAddress & ␦
55    strCity & strState & strZipCode & ")"
56    Set objDBConn = Server.CreateObject("ADODB.Connection")
57    objDBConn.Open "FILE NAME=d:\PubsDB.dsn"
58    objDBConn.Execute strSQLText
59    objDBConn.Close
60    %>
61    Store added to the database!
62    <HR SIZE=1 NOSHADE>
63    <P ALIGN="LEFT">
64    Based on the data from <U>form.htm</U>, this page inserts the values
65    into the database on a successful transaction.
66    </P>
67    </FONT>
68    </BODY>
69    </HTML>
```

בדף זה אין הרבה מה להסביר. שורות 18-23 מקבלות את נתוני הטופס מדף form.htm ומאחסנות אותם במשתנים מקומיים. בשורה 24, מבוצע חיתוך של מספר הטלפון המתקבל מהמשתמש (שורה 18) כך שמתקבל קוד ID מיוחד לשימוש בבסיס הנתונים. שורות 47-52 מעצבות את המשתנים המקומיים כך שייצרו את שאילתת SQL המיועדת להכנסת הערכים לבסיס הנתונים כרשומה. לבסוף, שורות 56-59 יוצרות חיבור (connection) לבסיס הנתונים ומפעילות את השאילתה, וכן סוגרות את החיבור לאחר ביצועה. בשורה 57, עליך להחליף את הנתיב לקובץ PubsDB.DSN כך שיצביע על קובץ DSN של בסיס הנתונים Pubs.

תוכל לבדוק את הדפים הללו על ידי פתיחת הדף form.htm בדפדפן האינטרנט שלך והזנת נתוני דוגמה. לאחר שתלחץ על לחצן Submit, אתה אמור לראות דף הדומה לזה שמוצג בתרשים 21.20.




**תרשים 21.20** לאחר עדכון מוצלח של בסיס הנתונים, אתה אמור לראות דף מעין זה.

כעת, נסה להזין שוב את אותם הנתונים על ידי לחיצה על לחצן reload/refresh שבדפדפן האינטרנט שלך. הפעם, תקבל הודעת שגיאה בתחתית הדף, שדומה לזו:

```
Microsoft OLE DB Provider for ODBC Drivers error '8004e14'
[Microsoft][ODBC SQ Server Driver][SQL Server]Violation of PRIMARY KEY constraint
'UPK_storeid'. Cannot insert duplicate key in object 'stores'.
/MCP.Projects/transaction.asp, line 70
```

שגיאה זו מציינת שהתסריט ניסה להזין ערך כפול לתוך בסיס הנתונים. ספציפית, הוא ניסה להכניס נתונים כפולים לעמודת מפתח עיקרי (Storeid). כאן מגיע השלב הבא: קביעה אם התנועה (Transaction) הצליחה או נכשלה.

## Commit or Abort

כפי שקראת קודם לכן, MTS מסווג תנועה כהצלחה (Commit) או ככישלון (Abort). תוכל לקבוע אם התנועה מוצלחת על פי שטף העבודה והלוגיקה שבתסריטים שלך, ובמידה והתנועה נכשלה, להצהיר על כך במפורש. לרוב, התסריטים שלך יסתמכו על תוצאות מתסריט או רכיב אחר כדי לקבוע את נתיב הפעולה הבאה. לדוגמה, התסריט

שלך עשוי לקרוא לרכיב בנקאות או לפונקציה כדי לבצע את התנאים המוקדמים לתנועה. אם התנועה הצליחה, הרכיב יחזיר ערך 0, ואם היא נכשלה, הוא יחזיר 1. לאחר שהתסריט שלך קבע שהפעולה לא הצליחה, הוא יכול להצהיר על התנועה כמבוטלת על ידי קריאה לשיטה ObjectContext.SetAbort.

### אזהרה:

מעשית, דפי תנועות תומכים רק בתנועות בסיסי נתונים, נכון להיום. אם התנועה נכשלת ותוצאותיה מבוטלות, MTS יבטל את השינויים. דבר זה לא חל ישירות על שינויי מערכת הקבצים או על משתני היישום\session. עם זאת, תוכל לעקוף ליקוי זה בעזרת תסריטים שכתובים היטב. מידע נוסף אודות נושא זה תקבל בהמשך פרק זה.

בדוגמה, כבר ראית מה יכול לגרום לתנועה להיכשל. כשאתה מנסה לשכפל נתונים בבסיס הנתונים, תקבל הודעת שגיאה אגרסיבית ונפוצה. תוכל להשתמש ב-MTS כדי לקבוע אם העדכון הצליח, ואם לא, להציג הודעת שגיאה ידידותית יותר. תוכל גם להשתמש ב-MTS כדי ללכוד מצב בו בסיס הנתונים אינו זמין, ולחסוך ממשתמשיך את המאמץ שבהזנת נתונים לבסיס נתונים לא זמין.

בדוגמה, תוכל להתבונן בכמה "אבני דרך" קטנות שניתן להתחשב בהן כנקודות "go/no go" בתנועה שלך. אבני דרך אלו יכולות להוות הזדמנות לבדיקת הצלחת התנועה ולביטולה אם יש צורך בכך. להלן שתי (אך לא בהכרח כל) ההזדמנויות בהן אתה יכול לבדוק אם יש בעיה בדף transaction.asp :

- שורות 18-24 של תדפיס 21.4 - הקוד בשורות אלו מניח שיש ערך שמתקבל מהטופס. אך מה יקרה אם הטופס לא יכיל נתונים או שהוא יכיל נתונים בפורמט לא נכון?
- שורה 62 - כאן מתבצעת שאילתת SQL ממשית אשר מכניסה את הנתונים לתוך בסיס הנתונים. מה יקרה אם הפעולה הזו תיכשל?

על ידי שימוש בתסריטים, אתה יכול לבנות בדיקות מותנות אל תוך הדף שלך כדי לקבוע אם התהליך מתבצע על פי המתוכנן. אם משהו משתבש, תוכל להכריז על ביטול התנועה ולבטל את התהליך. אם התהליך מתבצע כשורה, תוכל להניח לתנועה להתבצע עד לבדיקה הבאה.

בדוגמה זו, התחל בבדיקה אם התקבלו נתונים מהטופס. לרוב, תעדיף להציב את בדיקת הקלט בטופס עצמו. עם זאת, יש תמיד סיכוי שמשתמש הגיע לדף בטעות ללא הנתונים המתאימים. הוסף בדיקה מותנית עבור אפשרות זו :

**1.** פתח את הדף transaction.asp בעורך שלך.

**2.** אתר את שורות 18-24 בתדפיס 21.4 במקום בו מתקבלים שדות הטופס והחלף אותן בתדפיס 21.5.

**תדפיס 21.5** קביעה אם שדות הטופס מלאים

```
1  strStoreID = Request.Form("txtTelephone")
2  strName = Request.Form("txtName")
3  strAddress = Request.Form("txtAddress")
4  strCity = Request.Form("txtCity")
5  strState = Request.Form("txtState")
6  strZipCode = Request.Form("txtZipCode")
7  If strStoreID = "" or strName = "" or strAddress = "" or ↳
8  strCity = "" or strState = "" or strZipCode = "" Then
9        ObjectContext.SetAbort
10 Else
11       strStoreID = right(strStoreID, 4)
12 End If
```

**3.** שמור את השינויים בקובץ transaction.asp.

כאן, הכנס בדיקה פשוטה בשורות 7 ו-8 כדי לוודא ששדות הטופס אינם ריקים. אם הם ריקים, קרא לשיטת ObjectContext.SetAbort כדי להכריז על כישלון התנועה. אם שדות הטופס מכילים ערכים, המשך לשלב קיצוץ של מספר הטלפון לצורך יצירת קוד ID ייחודי לחנות.

לבסוף, בדוק אם שאילתת ה-SQL הושלמה בהצלחה. לשם כך, בדוק את מאפיין Count של אוסף Errors של האובייקט Connection. בדיקה פשוטה זו תגלה אם קרתה שגיאה בזמן ביצוע הפעולה בבסיס הנתונים. בצע את השלבים הבאים:

**1.** אתר את שורה 62 מתוך תדפיס 21.4 שבקובץ transaction.asp, במקום בו השאילתה מבוצעת. החלף את השורה בתדפיס 21.6.

**תדפיס 21.6** קביעה אם החיבור (connection) לבסיס הנתונים פעיל

```
If objDBConn.Errors.Count > 0 Then
      ObjectContext.SetAbort
End If
```

**2.** שמור את השינויים שביצעת בקובץ transaction.asp.

**הערה:**

סביר להניח שתרצה להשתמש בבדיקת שגיאות מתוחכמת יותר מזו שבדוגמה, אך דוגמה זו מסבירה את העניין בצורה טובה..

כעת, לאחר שהבדיקות המותנות שלך הושלמו, מה יקרה כשתפעיל את התסריט עם נתונים משוכפלים? לא יקרה הרבה. השלב הבא והסופי אותו עליך לבצע הוא הקוד המיועד לאירועי התנועות.

# כתיבת אירועי התנועות

השלב האחרון בדוגמת יישום התנועות הוא הוספת אירועי התנועות. ישנם שני אירועי תנועות אותם אתה יכול להגדיר בצורת שגרות:

- OnTransactionCommit - מופעלת אם התנועה הצליחה.
- OnTransactionAbort - מופעלת אם התנועה נכשלה, כפי שנקבע על ידי הבדיקות המותנות שלך או על ידי רכיב מותאם.

תוכל להשתמש באירועי התנועה כדי להפעיל תסריטים לאחר ביצועה, לאחזר מצב נתונים, או (כמו שבדוגמה שלנו) פשוט להציג הודעה למשתמש. כדי להשלים את הדוגמה, בצע את השלבים הבאים:

**1.** אתר את שורה 65 בתדפיס 21.4 שבדף transaction.asp שלך. זוהי השורה המציינת כי נתוני החנות הועלו בהצלחה. החלף את השורה הזו בתדפיס 21.7.

**תדפיס 21.7** הוספת אירועי התנועה

```
<%
Sub OnTransactionCommit()
      Response.Write("Your store has been successfully added to our
      ➥ database! Thank you!")
End Sub
Sub OnTransactionAbort()
      Response.Write("A problem occured when we tried to add your store
      ➥ to our database:<BR>")
      Response.Write("<B>Error:</B> Please <A HREF=""form.htm"">try
      ➥ again</A>.")
End Sub
%>
```

**2.** שמור את השינויים שביצעת בקובץ transaction.asp.

כשתנסה את הדפים כעת, אתה אמור לראות אחת משתי הודעות בתחתית דף transaction.asp, על פי התנועה. אם התנועה הצליחה ובסיס הנתונים עודכן, תראה את ההודעה הבאה:

Your store has been successfully added to our database! Thank you!

אך אם התנועה נכשלה בשל אחת מהבדיקות המותנות שלך, תראה הודעה כזו:

A problem occurred when we tried to add your store to our database:
Error: Please try again.

ברכותיי! יצרת את יישום האינטרנט מבוסס התנועות (Transactional) הראשון שלך!

# שימוש ברכיבי MTS בתוך ASP

בחלק השני והרביעי של ספר זה התנסית ביצירת רכיבים בצד השרת. סעיף זה מסתמך על הניסיון הזה לצורך יצירת רכיב MTS משלך בתוך דף ASP. בתחילת הפרק, כבר פיתחת רכיב והוספת אותו ל-MTS. כעת תעבור לשלב הבא והוא קריאה לרכיב זה מתוך דף ASP. ההבדל העיקרי בין צורת השימוש הזו ברכיב והצורה שנידונה בפרקים קודמים היא הצגת אירועי MTS.

כשפיתחת את ASP transactional שלך, הסתמכת על שני אירועים: OnTransactionCommit ו-OnTransactionAbort. בדוגמה שלנו נשתמש באותם שני האירועים בהבדל אחד: אירועי התנועה (Transaction) ייקראו מתוך הרכיב האישי שלך. זכור כי הרכיב שלך משווה שני ארגומנטים וקובע אם הם מתאימים. אם הם ייחודיים, התנועה תיחשב כמוצלחת. אם הם זהים, התנועה תיכשל. התחל זאת ביצירת דף HTML פשוט עם טופס שמיועד להתחלת התנועה:

1. צור דף אינטרנט חדש בשם component-form.htm.
2. הכנס את תדפיס 21.8 לתוך הדף החדש שלך.

**תדפיס 21.8** דף component-form.htm.

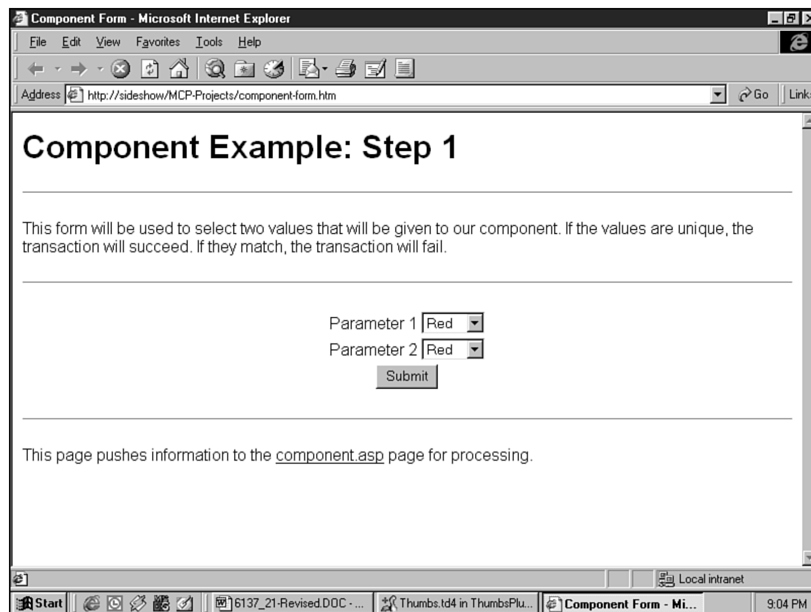
```
<HTML>
<HEAD>
<TITLE>Component Form</TITLE>
</HEAD>
<BODY>
<FONT FACE="Arial">
<H1>Component Example: Step 1</H1>
<HR SIZE=1 NOSHADE>
<P ALIGN="LEFT">
    This form will be used to select two values that will be given to
    our component. If the values are unique, the transaction will
    succeed. If they match, the transaction will fail.
</P>
```

```
<HR SIZE=1 NOSHADE>
<FORM action="component.asp" method=POST>
<TABLE BORDER="0" ALIGN="CENTER">
<TR>
<TD>
      Parameter 1
</TD>
<TD>
      <SELECT name=selParam1>
          <OPTION>Red</OPTION>
          <OPTION>Blue</OPTION>
          <OPTION>Green</OPTION>
      </SELECT>
</TD>
</TR>
<TR>
<TD>
      Parameter 2
</TD>
<TD>
      <SELECT name=selParam2>
          <OPTION>Red</OPTION>
          <OPTION>Blue</OPTION>
          <OPTION>Green</OPTION>
      </SELECT>
</TD>
</TR>
<TR>
<TD COLSPAN=2 ALIGN="CENTER">
      <INPUT type="submit" value="Submit">
</TD>
</TR>
</TABLE>
</FORM>
<HR SIZE=1 NOSHADE>
<P ALIGN="LEFT">
  This page pushes information to the <U>component.asp</U> page for
  processing.
</P>
</FONT>
</BODY>
</HTML>
```

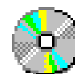
**הערה:**

ניתן למצוא את הדף component-form.htm בתקליטור המצורף.

הדף component-form.htm, כמו דוגמת ASP transactional, הוא טופס פשוט מאוד שידחוף נתונים לדף ASP. במקרה זה, אתה מסתמך על שתי תיבות נפתחות (selParam1 ו-selParam2) שיעבירו שני ערכים (צבעים). ערכים אלה נדחפים לדף component.asp, על פי הפעולה המוגדרת בשורה 15. דף זה מוצג בתרשים 21.21.




**תרשים 21.21** הדף component-form.htm מיועד לדחוף ערכי צבעים לרכיב.



השלב הבא הוא ליצור את הדף component.asp (שנמצא בתקליטור המצורף), אשר ייקח את הערכים שהתקבלו מהטופס ויעבירו אותם אל הרכיב MTS שלך. לשם כך, בצע את השלבים הבאים:

**.1** צור דף ASP חדש בעורך שלך וקרא לו בשם component.asp.

**.2** הכנס את תדפיס 21.9 לתוך הדף component.asp.

```
<%@TRANSACTION=Required Language=VBScript %>
<HTML>
<HEAD>
<TITLE>Component Example</TITLE>
</HEAD>
<BODY>
<FONT FACE="Arial">
<H1>Component Example: Step 2</H1>
<HR SIZE=1 NOSHADE>
<P ALIGN="LEFT">
This page takes the information provided by <U>form.htm</U> and
inserts it into the database for storage.
</P>
<HR SIZE=1 NOSHADE>
<%
      On Error Resume Next
      Dim strParam1, strParam2
      Dim objMTSValidate
  strParam1 = Request.Form("selParam1")
  strParam2 = Request.Form("selParam2")
  If strParam1 = "" or strParam2 = "" Then
          ObjectContext.SetAbort
  End If
%>
<TABLE ALIGN="CENTER">
<TR>
<TD>
      <B>Parameter 1</B>:
</TD>
<TD>
      <%= strParam1 %>
</TD>
</TR>
<TR>
<TD>
      <B>Parameter 2</B>:
</TD>
<TD>
      <%= strParam2 %>
</TD>
</TR>
</TABLE>
<HR SIZE=1 NOSHADE>
<%
Set objMTSValidate = Server.CreateObject("MTSValidate.ValidateData")
CheckParam = objMTSValidate.CheckParam(strParam1, strParam2)
%>
```

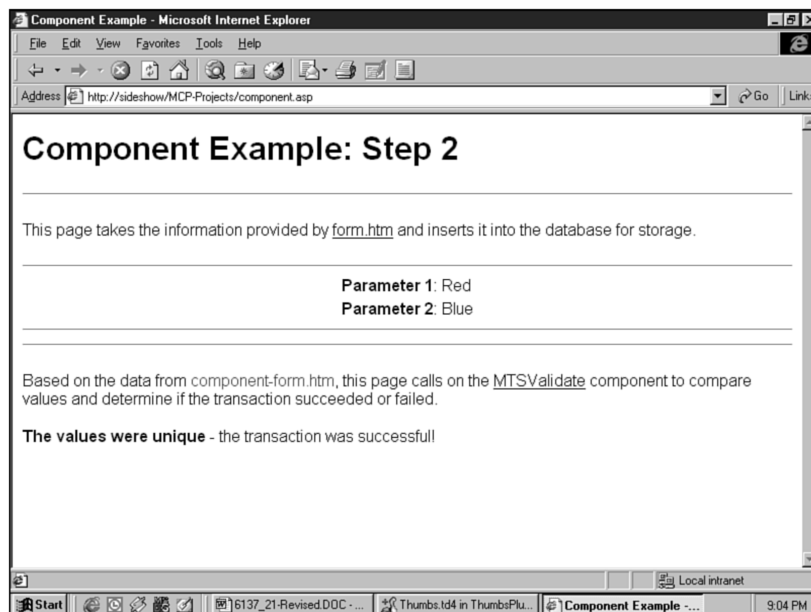
```
48  <%
49  Sub OnTransactionCommit()
50        Response.Write("<B>The values were unique</B> - the transaction
          ↳ was successful!")
51  End Sub
52  Sub OnTransactionAbort()
53        Response.Write("<B>The values were the same</B> - the transaction
          ↳ failed.")
54  End Sub
55  %>
56  <HR SIZE=1 NOSHADE>
57  <P ALIGN="LEFT">
58  Based on the data from <A HREF="component-form.htm">
59  component-form.htm</A>, this page
60  calls on the <U>MTSValidate</U> component to compare values and
61  determine if the transaction succeeded or failed.
62  </P>
63  </FONT>
64  </BODY>
65  </HTML>
```

**3.** שמור את השינויים בדף החדש שיצרת.

הבה נבחן מה קורה בדף component.asp. בשורה 1, אתה יוצר תנועה עבור הדף והרכיבים שלו. הרכיב לא ייצור תנועה (Transaction) חדשה, ולכן הוא ישתמש בתנועת הדף לצורך פעילותו. שורות 19-23 קוראות את ערכי הטופס מתוך הדף component-form.htm ומוודאות שערכים אלה אינם ריקים. אם אחד מהערכים ריק (למרות שלא סביר שדבר מעין זה יקרה), התנועה תבוטל אוטומטית על ידי קריאה ל-ObjectContext.SetAbort בשורה 22. אפשרות אחרת שיש לקחת אותה בחשבון היא שמשתמש ייגש לדף component.asp ישירות בלי שישתמש קודם לכן בטופס, במצב בו הערכים יישארו ריקים. עליך לצפות כל מצב אפשרי.

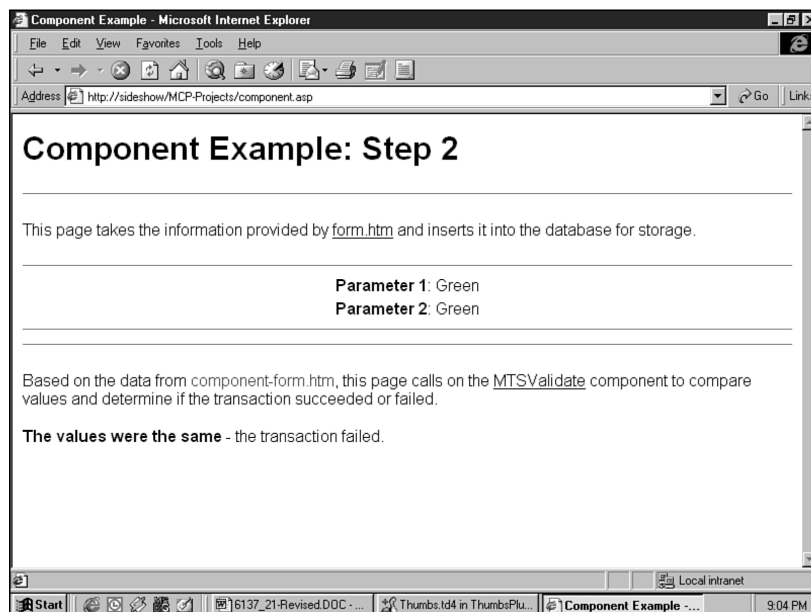
הפעולה האמיתית מתחילה בשורה 45, שם נוצר מופע (instance) חדש של רכיב MTS, שזהו למעשה הרכיב MTSValidate.DataValidate MTS שלך. שורה 46 קוראת לרכיב על ידי העברת שני ערכי הטופס כארגומנטים הנדרשים. הרכיב קורא פנימית את שני הערכים וקובע אם התנועה הצליחה (הערכים ייחודיים) או אם היא נכשלה (הערכים זהים). לאחר שהרכיב קיבל החלטה, הוא קורא לשיטת SetCommit או SetAbort על פי החלטתו. אירועים אלה "צפים" אל דף ה-ASP והם נקלטים (Intercepted) על ידי אירוע OnTransactionCommit בשורות 49-51 או אירוע OnTransactionAbort בשורות 52-54.

פתח את הדף component-form.htm בדפדפן האינטרנט וראה מה יקרה. אם התנועה הצליחה, יוצג דף כגון זה שבתרשים 21.22 בדפדפן האינטרנט שלך.




**תרשים 21.22** הערכים אינם מתאימים, ולכן הרכיב הכריז על התנועה כמוצלחת.

אך אם התנועה נכשלה, תראה דף שדומה לזה שמוצג בתרשים 21.23.




**תרשים 21.23** הפעם, ערכי הצבעים זהים ולכן הרכיב הכריז על כישלון התנועה.

שים לב כיצד הדף נטען במהירות לאחר השימוש הראשון. דבר זה קורה הודות להטמנת רכיבים ב-MTS, המאפשרת לדף להתייחס לתוכן האובייקט בניגוד להתייחסות למופע (instance) ממשי של רכיב, אותו יש ליצור ולמחוק. אם תשאיר את הדוגמה הזו כשלוש דקות ולא תבצע גישה לדף במשך זמן זה, MTS יהיה חייב להפעיל מחדש את הרכיבים.

# סיכום

Microsoft Transaction Server פותח עולם חדש של פיתוח רכיבים ויכולת הרחבה ליישומי האינטרנט שלך. על ידי שימוש ב-MTS, אתה יכול לשפר את ביצועי היישומים שלך, את יציבותם, ויכולת ההרחבה שלהם מבלי להסתמך על קידוד ידני מורכב. MTS הוא אבן הפינה של +COM החדש אשר הופיע לראשונה ב-Windows 2000 וסלל דרך מהפכנית ומצוינת ליישומי האינטרנט שלך. בפרק 22, תלמד כיצד לשלב שכבת גמישות נוספת וסבילות תקלות לתוך היישומים שלך.

## פרק 22

# שימוש ב-ASP עם MSMQ
# (Microsoft Message Queue Server)

**בפרק זה:**

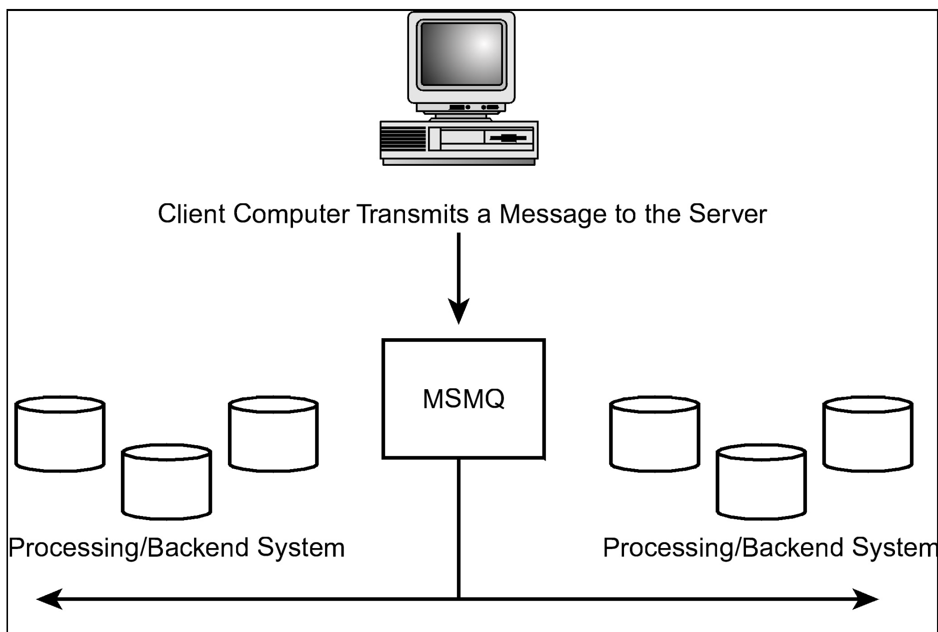
- **מדוע להשתמש ב-MSMQ?**
- **תקשורת ברשת עם MSMQ**
- **התקנה והגדרה של שירותי MSMQ**
- **שימוש בדפי ASP עם MSMQ**

בחיי יישום אינטרנט נוצר לעיתים מצב בו הוא חייב להתקשר עם מערכות אחרות דרך רשת. בעבר, התקשורת במערכות מבוזרות אלו הסתמכה על fault tolerance מתוכנתת. Microsoft Message Queuing Services (MSMQ) מאפשר ליישומים שלך להתקשר ביניהם על פני רשתות ומערכות הטרוגניות, גם אם הן אינן מקוונות זמנית. על ידי שימוש ב-MSMQ, היישומים שלך לא ייאלצו להמתין עד אין סוף לתגובה ממערכת שייתכן והיא אינה זמינה. לחילופין, אתה יכול להשתמש ב-MSMQ כדי לעבד דרישות תקשורת ולהמשיך לשלב החשוב הבא. MSMQ מתאים במיוחד ליישומים שפועלים בצורה דומה לדרך פעולת הדואר האלקטרוני - אתה רוצה שההודעה תגיע, אך אין צורך שההודעה תגיע דווקא ברגע בו שלחת אותו. לדוגמה, אתה יכול להשתמש ב-MSMQ בתוך דף ASP כדי להתקשר עם שרת בסיס נתונים מרוחק, מערכת חיוב רחוקה, או אפילו יישום לוח זמנים ישן.

# מדוע להשתמש ב-MSMQ?

MSMQ הוא ערכה של שירותים בעזרתם אתה יכול להוסיף fault tolerance ויכולת הרחבה (Scalability) לתקשורת של יישומי האינטרנט שלך. למרות שאין צורך בכך עבור MSMQ, השילוב של MTS ו-MSMQ נותן גמישות ואמינות רבים יותר ליישומי האינטרנט שלך כשאתה עוסק בתקשורת. MTS מטפל בשאלות "האם פעולה זו הצליחה או נכשלה?", ו-MSMQ מטפל בשאלה "האם דרישה זו התקבלה על ידי השירות המתאים?". כשמדובר ביישום אינטרנט גדול מאוד או כזה שברמת הארגון, לרוב שרת האינטרנט מתקשר עם מקור נתונים אחד או יותר, כגון בסיס נתונים או מערכת עיבוד מותאמת. בכל פעם שיישום האינטרנט מתקשר עם בסיס הנתונים או עם מערכת העיבוד, נשלחת דרישה ויישום האינטרנט ממתין לתגובה. אם, מכל סיבה שהיא, המערכת או בסיס הנתונים אינם זמינים, יישום האינטרנט נתקל במכשול שהוא אינו יכול להתגבר עליו והוא מפסיק לפעול.

MSMQ מציע רמה חדשה של fault tolerance ליישומי אינטרנט, אשר היתה מוכרת רק לשרתים עסקיים בקנה מידה גדול ולמחשבי Mainframes (כגון מערכות עסקאות פיננסיות). על ידי שימוש ב-MSMQ, יישומי האינטרנט אינם זקוקים יותר לנהל ישירות את התקשורת עם מערכות אחרות, אלא הם משתמשים בקריאות לשירותי ההודעות של MSMQ כדי לנהל את פרטי המשלוח וקבלת ההודעות למערכת המצוינת. כעת כל שנותר ליישום שלך הוא לדאוג ליצירת ההודעה שתישלח ולהבין את ההודעה שמתקבלת. כל זמן שכל היישומים שלך דוברים אותה שפה, הם יכולים להשתמש ב-MSMQ עבור התקשורת שלהם. תרשים 21.1 מדגים את הרעיון.



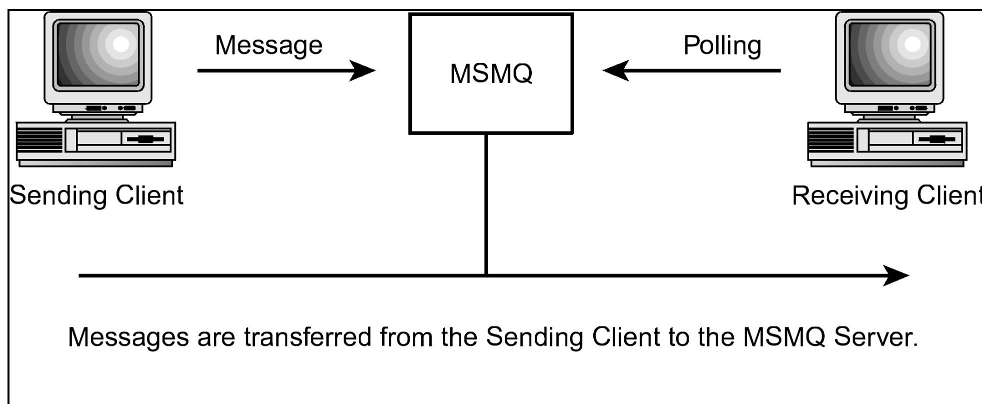

**תרשים 22.1** MSMQ מסיר מהיישום שלך את הצורך בטיפול בתקשורת בין מערכות.

# משלוח וקבלת יישומים

בעולם MSMQ, ישנם שני סוגי יישומים: שולח ומקבל. ניתן לראות ביישום שולח כיוזם תנועת הודעות. היישום השולח מתחיל את תהליך התקשורת על ידי העברת הודעה לשירותי MSMQ. הודעה זו מתקבלת על ידי היישום המקבל לצורך עיבוד, לאחר שהוא גילה שיש לו הודעה שממתינה עבורו. תוכל לחשוב על התהליך במונחים של דואר אנושי:

1. אתה שולח מכתב. אתה, בתור שולח ההודעה, מכין את ההודעה שלך (״היי, מה נשמע?״) וסוגר את המעטפה.
2. אתה שולח את ההודעה דרך שירות הדואר. שירות הדואר, המשמש כנושא ההודעה, מטפל בפרטי הבאת המכתב מהמיקום שלך אל הנמען. זהו תפקיד MSMQ.
3. הנמען, שיודע שמגיע אליו דואר חדש כל יום בשעה קבועה, בודק בשעה זו אם הגיע דואר. הוא רואה את המכתב שלך על מפתן דלתו ולוקח אותו. הוא פותח את המעטפה, קורא את ההודעה, ומתייחס אליה.

MSMQ אחראי להעביר את ההודעות האלקטרוניות של היישום אל היישום המקבל. הרעיון אותו יש להבין הוא שהיישום המקבל את ההודעה חייב לבדוק אם יש לו הודעות כל כמה זמן, אחרת ההודעה לא תיקרא במשך זמן ארוך. תרשים 22.2 מדגים את רעיון היישומים השולחים והמקבלים.




**תרשים 22.2** משלוח וקבלה ביישומים דומה לפעולת שירות הדואר בעולם האמיתי.

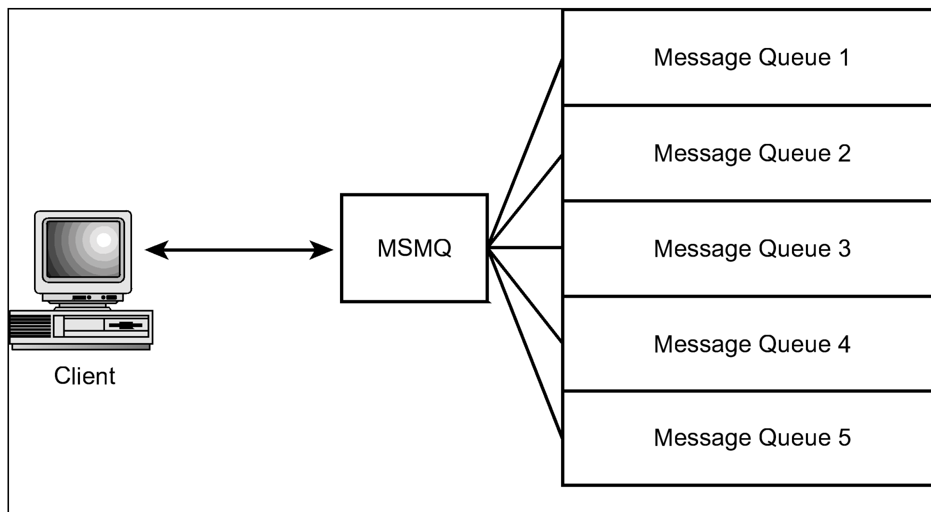
MSMQ מסוגל גם להגדיר להודעות עדיפויות ומאפיינים. היישום שלך יכול לתת חשיבות רבה יותר לתקשורות מסוימות. על ידי שימוש ב-MSMQ, אתה יכול להקצות מאפיינים להודעות שלך, אשר ינחו את היישומים המקבלים לטפל בהן בצורה שונה. ניתן להתייחס להודעות בצורה שונות גם בתוך MSMQ עצמו ובצורת העברת הנתונים שלו. בדוגמת שירות הדואר, יש לך אפשרות לשלוח את המכתב שלך בדואר רגיל, אקספרס, או רשום. MSMQ מספק ליישומים שלך פונקציונליות דומה למדי כשהעברה פשוטה של הודעה אינה מספיקה.

# תורי הודעות

אם לא היה סדר כלשהו מאחורי העניין, MSMQ היה הופך לערב רב של הודעות המגיעות מיישומים שולחים, דבר שהיה יוצר מערכת העברות מבולגנת כלפי היישומים המקבלים אותן. כדי לסדר את העניינים, MSMQ משתמש ברעיון **תורי ההודעות** (Message Queues). תור הודעות דומה לערוץ בטלוויזיה - לכל ערוץ יש את התוכן הייחודי שלו ויש הבדל בינו לבין הערוצים האחרים. כמפתח, אתה יוצר תורי הודעות שמיועדים לערוצי תקשורת משותפים בין היישומים השולחים ליישומים המקבלים. תורים אלה מהווים את התשתית המשותפת בין שני היישומים. תורי ההודעות מאוחסנים בתוך בסיס הנתונים של MSMQ ב-Microsoft SQL Server בשרתי MSMQ ובאחסון נתונים פנימי אצל הלקוח (פרטים נוספים על כך בהמשך).

תורי ההודעות הם קבוצות הודעות נפרדות המסווגות על פי צרכי היישום שלך. לדוגמה, היישום עשוי לדרוש סדר מסוים של מאגר ההודעות הנכנסות וסדר אחר של התגובות שלו עצמו. ניתן ליישם כל אחד ממאגרים אלה כתור הודעות, ובכך לאפשר ליישום שלך להתמקד בתורים ספציפיים כשהוא מחפש סוגי הודעות מסוימים. החלופה לכך היא להשתמש במאגר הודעות מרכזי אחד אותו היישום חייב לסרוק כדי לאתר את סוג ההודעה לה הוא מצפה.

אין מגבלות ממשיות על מספר תורי ההודעות שאתה יכול ליצור בתוך MSMQ, תוכל להגדיר תורי הודעות נפרדים עבור התהליכים השונים שביישום שלך, ולהבטיח את שלמות התקשורת. כחלק מתהליך העברת וקבלת ההודעות, היישומים שלך מציינים באיזה תור הודעות יש להשתמש. תרשים 22.3 מדגים תורי הודעות ואת הקשר שלהם ליישומי משלוח וקבלה.




**תרשים 22.3** תורי הודעות הם ערוצים עצמאיים לקיום תקשורת בין יישומים.

# תקשורת ברשת עם MSMQ

היתרון החשוב בשימוש ב-MSMQ בשגרות התקשורת שביישומים שלך הוא יציבות התהליך. MSMQ מסוגל לנהל הודעות על פני רשתות ושרתים מרובים ולהעביר אליהם את ההודעה שלך. בכך, MSMQ פותח את יישומי האינטרנט שלך לעולם המחשוב המבוזר. לדוגמה, קח מערכת הזמנות בחברת תעופה. שירות הלקוחות של החברה וצוות המכירות יכולים לבצע הזמנות ולתכנן טיולים תוך שימוש ביישום אינטרנט ובדפדפן אינטרנט סטנדרטי. ניתן להעלות את ההוראות לשרת יישום האינטרנט על ידי שימוש ברכיבי COM שמנוהלים על ידי Microsoft Transaction Server (למידע נוסף אודות MTS, פנה לפרק 21). רכיבים אלה, שמשמשים כשרת היישום, עשויים לדרוש עיבוד שמתבצע על ידי מערכת חיצונית, כגון בסיס נתונים. בשלב זה, MSMQ נכנס לזירה.

בהנחה שיש יישום בשרת המרוחק אשר ממתין לקבלת הודעות (כגון רכיב מותאם או מערכת עיבוד חכמה), MSMQ יכול לוודא שההודעה התקבלה. שרת יישום האינטרנט, המשתמש באובייקטי MSMQ, יכול לשדר הודעה לתור הודעות מסוים (לדוגמה, תור של הזמנות). MSMQ מטפל בפרטי העברת ההודעה על גבי הרשת, ומאפשר ליישום האינטרנט להמשיך לשלב הבא. לאחר שמערכת העיבוד קיבלה את ההודעה דרך MSMQ, היא יכולה לשדר תשובה למערכת יישום האינטרנט דרך תור הודעות שונה.

שרת יישום האינטרנט, המשגיח על התור וממתין להודעות חדשות, מודיע לנציג שירות הלקוחות שהגיעה הודעה חדשה (לדוגמה, אישור הזמנה).

**טיפ:**

לא כל יישום מסוגל לפעול בצורה טובה בסביבה מהסוג הזה. אם יישום האינטרנט שלך מסתמך על קבלת נתונים מיידית, אתה יכול עדיין להסתמך על אמינות העברת ההודעות של MSMQ. עם זאת, עדיין עליך להוסיף שגרות fault tolerance במקרה של עומס או תקלה. אם היישום שלך מסוגל לשרוד גם בלי קבלת משוב מיידי מהמערכת המרוחקת, MSMQ ייתן לך יותר חופשיות. הדבר תלוי לחלוטין בארכיטקטורת יישום האינטרנט שלך וצרכיו.

# הבטחת ההעברה

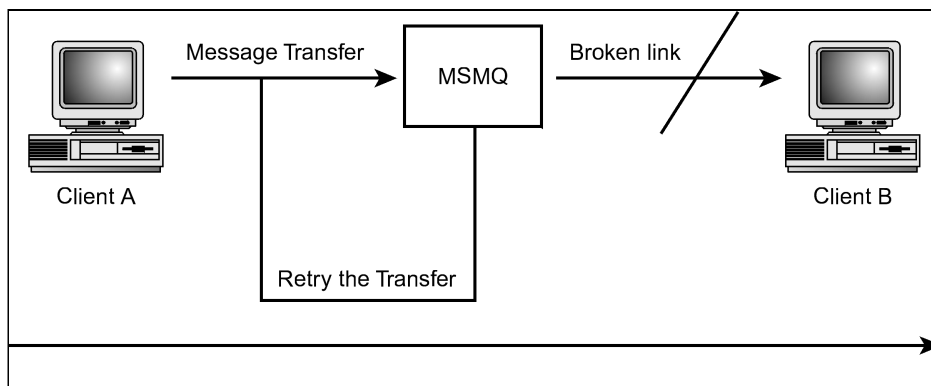
ל-MSMQ יש תכונה חשובה של **הבטחת ההעברה** (Delivery Assurance). מכמה בחינות, MSMQ הוא שרת תנועות (Transaction server). הוא מזהה כל הודעה לגופה בשירותי התורים שלו ומוודא שהיא עברה. אם ההודעה לא משודרת כראוי ללקוח MSMQ המרוחק, היא נשלחת בשנית. MSMQ פועל על העיקרון שההודעה חייבת להתקבל במלואה וכן חייב להתקבל אישור מסירה כדי שההעברה תיחשב כמוצלחת. דבר זה מגן על המערכת מפני אובדן או שכפול הודעות. במילים פשוטות, MSMQ קובע אם ההודעה הגיעה ליעדה או לא - ללא אפשרויות ביניים.

MSMQ משתמש בשתי שיטות להעברת הודעות: **אקספרס** ו**ניתן להתאוששות** (Recoverable). בעיקרון, העברת אקספרס היא הדרך הסטנדרטית למשלוח הודעות. שיטה זו מסתמכת על כך שהלקוח מקבל את ההודעה במשלוח הראשון - כלומר, MSMQ אינו מאחסן את תוכן ההודעה למקרה שיהיה צורך במשלוח חוזר. דרך זו היא היעילה ביותר להעברת הודעות משום שהיא דורשת פחות משאבים. מצד שני, השיטה שניתנת להתאוששות, דורשת יותר משאבים משיטת האקספרס. שיטה זו מוסיפה שכבת הגנה הדרושה על ידי יישומים רבים. עותק של ההודעה מאוחסן בשרת MSMQ למקרה שיהיה צורך לשחזר אותה במצב של כישלון בהעברתה. תרשים 22.4 מציג את הרעיון.

# שרתי MSMQ

מאחורי הקלעים, MSMQ מסתמך על שרתים. ישנם כמה סוגי שרתי MSMQ:

- PEC - (PEC) Primary Enterprise Controller מקביל ל-Primary Domain Controller (PDC) ברשת Windows NT. בכל רשת בה נעשה שימוש ב-MSMQ, חייב להימצא שרת PEC. שרת זה מתפקד כמנהל המרכזי של תעודות האישורים (Certification Keys) בהם משתמשים ב-MSMQ כדי לאמת הודעות. PEC מתפקד גם כשרת ניתוב MSMQ.




**תרשים 22.4** MSMQ: כשההודעה חייבת, ללא ספק, לעבור על גבי רשת

**הערה:**

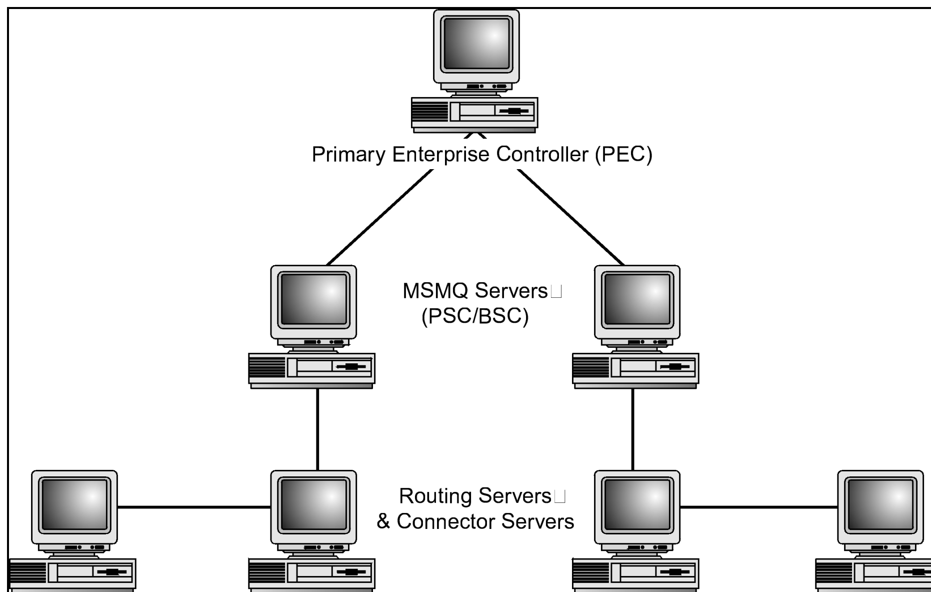
MSMQ תומך בתעודות אישורים בצורה דומה למדי לשרת נתונים באינטרנט. על ידי שימוש בתעודות דיגיטליות אלו, ניתן לחתום דיגיטלית על הודעות כדי להבטיח את תקפות ההודעה. בפרק זה, אינני דן בתעודות דיגיטליות, אך אם אתה מעוניין ללמוד עוד על הנושא, תוכל לקרוא את סעיף Microsoft Message Queue Server בתיעוד Windows NT 4.0 Option Pack.

- Primary Site Controller (PSC) - כל אתר נוסף, שמהווה קבוצה לוגית של לקוחות ושרתי MSMQ מחוץ ל-PEC שלך, דורש PSC. PSC מתפקד גם כשרת ניתוב MSMQ והוא מכיל עותק של כל התורים והמחשבים שהם חלק מהרשת.
- Backup Site Controller (BSC) - מקביל ל-Backup Domain Controller (BDC) ברשת Windows NT. BSC מספק שכבת כפילות עבור PEC ו-PSC במקרה של נפילת אחד מהם. BSC מכיל עותק לקריאה בלבד של כל נתוני המחשבים והתורים והוא יכול לתפקד כשרת ניתוב MSMQ.
- MSMQ Routing Server - שרתים אלה משמשים לתמיכה בניתוב דינמי של הודעות וכן כחוצץ לקידום הודעות. על ידי שימוש בשרת ניתוב MSMQ, הרשתות שלך יכולות להשתמש בפרוטוקולים שונים לצורך קישור לקוחות שונים ברשת.
- MSMQ Connector Server - כל שרתי MSMQ יכולים לתפקד כשרת חיבור MSMQ. שרתי חיבור MSMQ מאפשרים ליישומים מבוססי MSMQ להתקשר עם לקוחות שאינם מבוססי MSMQ, כגון שירות תור הודעות חיצוני. כדי לפתח שרת חיבור MSMQ, יש צורך בערכת SDK של MSMQ.

**הערה:**

עבור כל שרת MSMQ, יש צורך בבסיס נתונים בשרת SQL כדי לאחסן את כל נתוני הרשת וההודעות.

כל סוג שרת MSMQ מעביר נתונים למחשבים שונים ברשת שלך. תרשים 22.5 מדגים את השימושים בסוגים השונים של שרתי MSMQ.




**תרשים 22.5** Primary Enterprise Controller של MSMQ מתאם בין כל שרתי MSMQ.

**הערה:**

בהמשך הפרק, אדון בהתקנת MSMQ.

# לקוחות עצמאיים ותלויים

שרת MSMQ מטפל באחסון והעברת ההודעות בתורים, ומשום כך לקוחות MSMQ הם משתמשי הקצה של המערכת. יש להגדיר כל מחשב שיבצע פעולות יחד עם MSMQ כלקוח MSMQ. MSMQ תומך בשני סוגי לקוחות:

- לקוחות עצמאיים - לקוחות אלה משתמשים במלוא העוצמה והגמישות המוצעות על ידי MSMQ. לקוחות עצמאיים מנהלים את אחסון ההודעות של עצמם והם מסוגלים ליצור תור הודעות משלהם אם הם אינם מסוגלים להתקשר עם שרת MSMQ לצורך העברתן. לאחר ששרת MSMQ חוזר להיות זמין, תור ההודעות המקומי מועבר אליו. לקוחות עצמאיים מתוכננים ל-fault tolerance גבוהה והם מבטיחים שההודעות יעברו.

- לקוחות תלויים - אלה הם לקוחות MSMQ "טיפשיים" ללא יכולת אחסון מקומי. לקוחות אלה תלויים בשרתי MSMQ לצורך העברת וקבלת הודעות. אם השרת אינו זמין, כל הודעה שנשלחה תאבד.

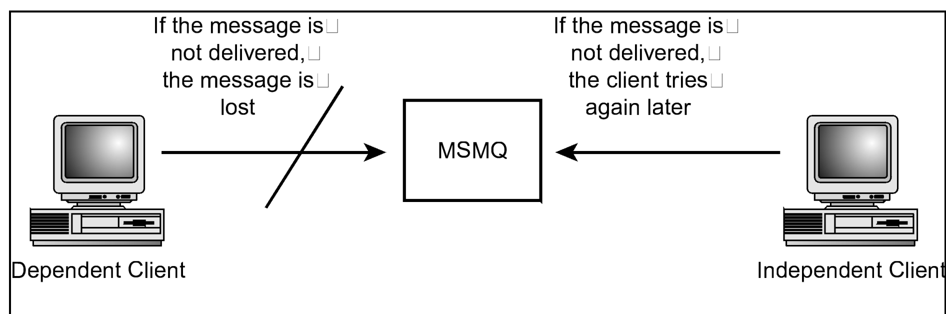
### הערה:

מדוע להשתמש בלקוח תלוי? יש לכך כמה סיבות. ראשית, אם הודעות היישום שלך אינן קריטיות, אין צורך בהוספת תקורת ניהול לקוח עצמאי. שנית, אם היישום שלך משתמש בתכונות fault tolerance משל עצמו, ייתכן ולא תזדקק לתכונות הלקוח העצמאי אשר מבצע ניסיונות משלוח חוזרים משל עצמו.

### טיפ:

דף ASP שלך יכול להיות לקוח עצמאי או תלוי. דבר זה נקבע בהתקנת MSMQ בשרת האינטרנט שלך. מידע נוסף אודות התקנות תקבל בהמשך הפרק.

תרשים 22.6 מדגים את ההבדלים בין לקוח עצמאי ללקוח תלוי.




**תרשים 22.6** לקוח עצמאי מטפל באחסון הנתונים של עצמו, ואילו לקוחות תלויים נתונים לחסדי שרתי MSMQ.

## Public, Private, and Transactional Queues

כפי שלמדת קודם לכן, תורי ההודעות דומים מאוד לערוץ טלוויזיה. לערוצים נפרדים אלה יש מאפיינים ייחודיים. MSMQ, הפועל דרך ASP, תומך בשלושה סוגים של תורי הודעות, כשכל אחד מהם מיועד למטרה שונה:

- תור ציבורי - כל לקוחות MSMQ יכולים לגשת לתור ציבורי. תור זה מוצג ב-Active Directory של כל שרתי MSMQ, אותם יכולים לראות כל הלקוחות.
- תור פרטי - כשאתה זקוק לתור שלא יהיה גלוי לציבור כל לקוחות MSMQ, אתה יכול להשתמש בתורים פרטיים. לתורים אלה אין הגבלות אבטחה, אלא הם פשוט נסתרים מכל הלקוחות. משמעות הדבר היא שעליך לדעת את שם התור כדי לגשת אליו. התור אינו מופיע ב-Active Directory, ולכן זהו גם יתרון בביצועים.
- Transactional queue - כשאתה עובד עם MTS ותנועות (Transaction), סביר להניח שתור התנועות יהיה הבחירה הטובה ביותר שלך. תור התנועות מכיל הודעות תנועה, שלא ניתן לשלוח אותן על ידי שימוש בתור הודעות רגיל. למרות שאין צורך ב-MTS כדי להשתמש בתורי תנועות (MSMQ מסתמך על DTC לצורך ניהולם), ניתן להשתמש בהם בתיאום כדי להרחיב את תנועות MTS שלך.

**הערה:**

ניתן להגיע לסוג רביעי של תור הודעות, שהוא תור הודעות פרטיות בלבד, על ידי שימוש ב-Visual Basic וב-++Visual C.

# מבנה ההודעה

ההודעות שב-MSMQ אינן מורכבות. רוב ההודעות מורכבות מטקסט פשוט, ומאפשרות בכך קלות ופשטות של ההעברה. ההודעות מחולקות לשני חלקים נפרדים:

- תווית ההודעה - התווית מתפקדת בצורה דומה למדי לכותרת דף אינטרנט או מסמך. התווית מיועדת לזיהוי ההודעה והיא יכולה להכיל טקסט מיוחד שיציין את סוג ההודעה הנשלחת.
- גוף ההודעה - זהו לב תוכן ההודעה עצמה. חלק זה מרכיב, לדוגמה, את גוף המכתב. ההודעה אותה אתה שולח דרך MSMQ נמצאת לרוב בגוף, והיא משתמשת בתווית כמזהה.

**הערה:**

MSMQ תומך בכמה סוגים של תכני הודעות, החל ממחרוזות פשוטות וערכים מספריים וכלה באובייקטים, מערכים, וקבצים. בפרק זה, אדגים את עקרונות הודעות המחרוזת. למידע נוסף אודות סוגי נתונים נוספים, פנה לתיעוד MSMQ.

# התקנה והגדרה של שירותי MSMQ

MSMQ מתפקד כחלק ממערכת ההפעלה Windows NT, והוא מופיע כשירות הזמין ליישומים. התקנת MSMQ אינה מסובכת. עם זאת, יש להבין את האפשרויות הכלולות בה. ניתן לבצע התקנת MSMQ על שרתי MSMQ או על הלקוחות, כפי שהסברתי בתחילת הפרק.

## דרישות MSMQ

MSMQ זמין בשתי צורות: סטנדרטי וארגוני. המהדורה הסטנדרטית של MSMQ זמינה כחלק מ-Windows NT 4.0 Option Pack. הגירסה הארגונית היא חלק מ-Windows NT 4.0 Enterprise Edition. גרסת הארגון של MSMQ כוללת את תכונות MSMQ הבסיסיות וכן ארבע יכולות נוספות:

- אין הגבלה על מספר המשתמשים המחוברים ל-MSMQ.
- יכולת להתחבר למערכות תורי הודעות של יצרנים אחרים, כגון MQSeries של IBM.
- ניתוב אוטומטי לעקיפת רשתות לא פעילות בסביבה מרובת רשתות.
- ניתוב יעיל ואוטומטי בין שרתים כשישנן כמה רשתות זמינות.

פרק זה מתמקד במהדורה הסטנדרטית של MSMQ. אם אתה מעוניין במידע נוסף על גרסת הארגון, בקר בדף האינטרנט Message Queue Overview and Resources בכתובת http://www.microsoft.com/ntserver/appservice/exec/overview/MSMQ_Overview.asp

## הדרישות עבור שרתי MSMQ

בתחילת פרק זה קראת אודות סוגי שרתי MSMQ השונים. כדי להתקין שרת MSMQ, עליך להשתמש ב-Windows NT 4.0 Server או Windows NT 4.0 Server Enterprise Edition.

עבור כל רשת ארגונית יש צורך לפחות בשרת בקרה ארגוני ראשי (PEC) אחד. שרת זה מאחסן את העותק הראשי של מאגר נתוני MSMQ או בסיס הנתונים. עבור כל שרתי MSMQ, נדרש בסיס נתונים מסוג Microsoft SQL Server גירסה 6.5 ומעלה.

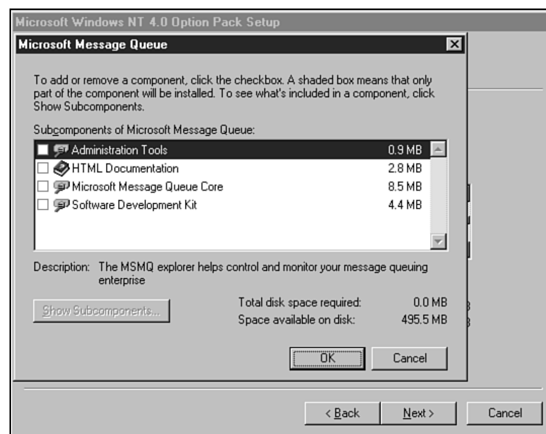
# דרישות עבור לקוחות MSMQ

כל מערכת בה מותקנת Windows 9x או Windows NT יכולה להיות מותקנת כלקוח MSMQ. כפי שציינתי קודם לכן, ישנם שני סוגי לקוחות MSMQ : עצמאי ותלוי. לצורך ביצועים טובים, עליך לוודא שכל מערכת שעשויה לא לקבל חיבור רשת לשרת MSMQ תותקן כלקוח עצמאי.

# התקנת שרת MSMQ

התקנת שרת MSMQ מתבצעת ב-Windows NT 4.0 Option Pack. עבור כל רשת ארגונית, יש צורך בלפחות בקר ארגוני ראשי אחד, ולכן זהו השלב הראשון בהתקנה. ניתן להוסיף עוד שרתים כגון בקרי אתר ראשיים (PSC), בקרי גיבוי אתר, או שרתי MSMQ עצמאיים. כדי להתקין PEC, בצע את השלבים הבאים :

1. פתח את תיבת ההגדרות של Windows NT 4.0 Option Pack ובחר באפשרות Add/Remove Components.
2. אתר את תיבת הסימון Microsoft Message Queue בתיבת הרשימה Component, ובחר בה.
3. לחץ על לחצן Show Subcomponents כדי לפתוח את תיבת הדו-שיח Microsoft Message Queue, המוצגת בתרשים 22.7.




**תרשים 22.7** כל רכיבי MSMQ מותקנים מתוך תיבת ההגדרות של Windows NT 4.0 Option Pack.

4. סמן את כל תיבות הסימון : Administration Tools, HTML Documentation, Microsoft Message Queue Core, ו-Software Development Kit.

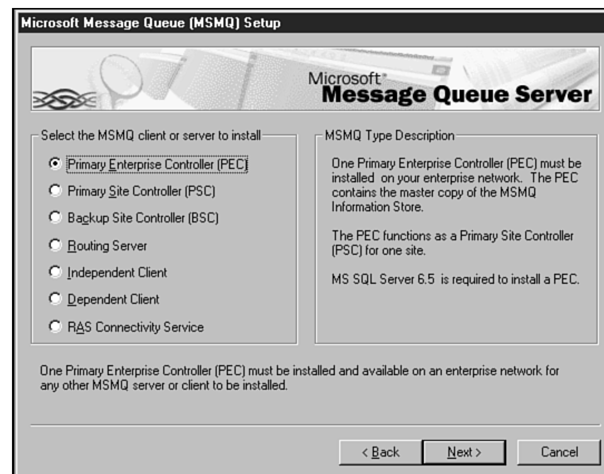
**הערה:**

SDK כולל גם מידע וקבצים בהם אתה יכול להשתמש במידה ואתה מפתח עבור MSMQ תוך שימוש ב-Visual Basic, Visual C++ או שפה גבוהה אחרת.

**5.** לחץ OK כדי להחיל את השינויים ולחזור לתיבת רשימת הרכיבים.

**6.** לחץ OK כדי לחזור לתיבת הדו-שיח Windows NT 4.0 Option Pack.

**7.** לחץ על לחצן Next כדי להתקדם למסך הבא, המוצג בתרשים 22.8.




**תרשים 22.8** לאחר שהגדרת את PEC, אתה יכול להתקין שרתי MSMQ אחרים ברשת שלך.

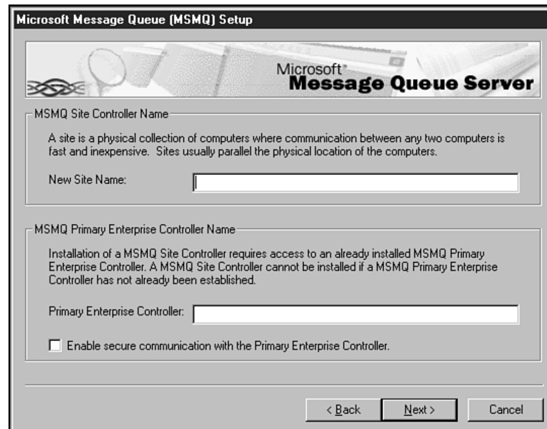
**8.** במסך זה, אתה יכול לבחור את סוג השרת (או הלקוח) אותו אתה מעוניין להתקין. בחר בלחצן האפשרויות Primary Enterprise Controller (PEC).

**9.** לחץ על לחצן Next כדי להגדיר את PEC אותו אתה מתקין. כעת יופיע הדף שמוצג בתרשים 22.9.

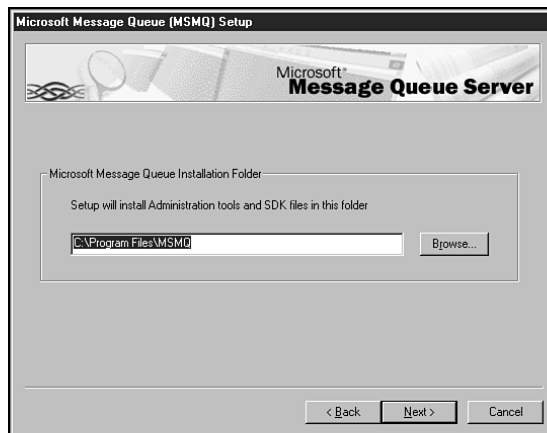
**10.** בלוח Primary Enterprise Controller Name, הקלד את שם הארגון שלך. שם זה יכול להיות שם קבוצת המחשבים של Windows NT או שם אחר שייצג את השרת.

**11.** בלוח Site Controller Name, הקלד את שם אתר MSMQ שלך. לרוב, זוהי רשת מקומית או קבוצת מחשבים.

**12.** לחץ על לחצן Next כדי להתקדם למסך הבא המוצג בתרשים 22.10.




**תרשים 22.9** התקנת PEC יוצרת מאגר נתוני MSMQ ראשי ובקר אתר ראשי.



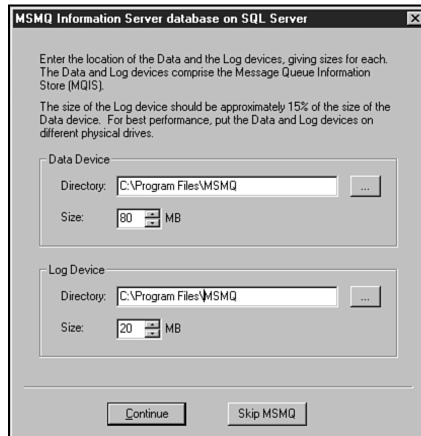

**תרשים 22.10** השתמש בתיבת דו-שיח זו כדי לציין את תיקיית ההתקנה של קבצים ב-MSMQ.

**13.** בשלב זה, ציין את נתיב ההתקנה עבור קבצי MSMQ. אתה יכול ללחוץ על לחצן Browse כדי לבחור תיקיה קיימת.

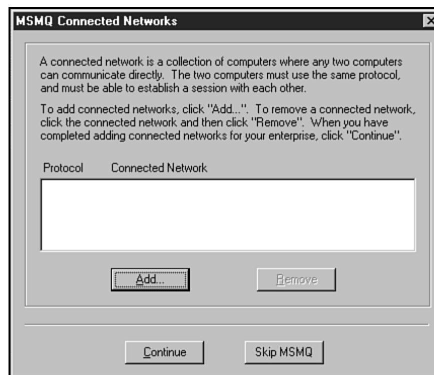
**14.** לחץ על לחצן Next כדי להתחיל את ההתקנה. תוכנית ההתקנה של Windows NT 4.0 Option Pack תתחיל להעתיק את קבצי MSMQ למערכת שלך. לאחר השלמת תהליך ההעתקה, תופיע תיבת הדו-שיח המוצגת בתרשים 22.11.

**15.** בנקודה זו, אתה יכול לציין מיקומי תיקיות בהם יאוחסנו בסיס הנתונים של MSMQ וקבצי Log של השרת. בנוסף, אתה יכול לציין גדלים התחלתיים של שני בסיסי הנתונים. לחץ על לחצן Continue כדי לאשר את ערכי ברירות המחדל.

**16.** כעת תופיע תיבת הדו-שיח MSMQ Connected Networks, המוצגת בתרשים 22.12. תיבת דו-שיח זו מיועדת לציון שמות הרשתות אליהן השרת שלך מחובר. לחץ על לחצן Add כדי להמשיך.

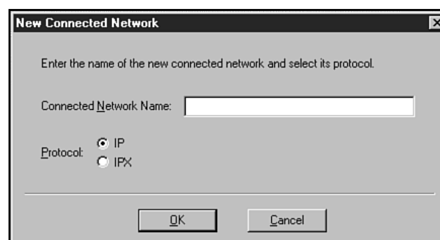



**תרשים 22.11** תיבת דו-שיח זו מציינת את גודל בסיס הנתונים לצורך אחסון נתוני MSMQ.




**תרשים 22.12** עליך לספק שם רשת עבור כל רשת המוגדרת ברשת שלך.

**.17** תיבת הדו-שיח New Connected Network, המוצגת בתרשים 22.13, מיועדת להוספת שם רשת להתקנת MSMQ. הקלד שם עבור הרשת שלך בשדה Connected Network Name. לדוגמה, אם הרשת שלך מקיפה את משרד המכירות, אתה יכול להקליד את השם Sales Office.



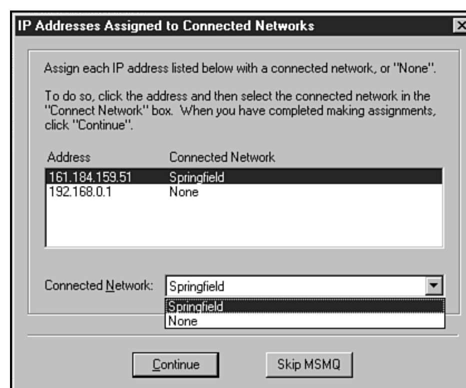

**תרשים 22.13** אתה יכול לציין שם רשת ופרוטוקול.

**18.** ציין את הפרוטוקול בו הרשת שלך משתמשת : IP או IPX. בחר בלחצן האפשרויות של הפרוטוקול המתאים.

**19.** לחץ OK כדי להמשיך.

**20.** לאחר שסיימת להוסיף שמות רשת עבור כל רשת מותקנת, לחץ על לחצן Continue. השלב הבא הוא להקצות שמות רשת לרשתות המותקנות.

**21.** אם יש לך כמה רשתות IP המוגדרות בשרת שלך, תופיע תיבת הדו-שיח IP Address Assigned to Connected Networks (ראה תרשים 22.14). בחר ברשתות שלך מתוך הלוח העליון ובחר בשם הרשת מתוך התיבה הנפתחת Connected Network.




**תרשים 22.14** שמות הרשת מיועדים להתייחסות לרשתות המחוברות פיסית לשרת שלך.

## הערה:

כל כתובת IP (הכתובת המזהה את המחשב) הקשורה לכרטיסי ממשק השרת שלך תופיע ברשימה זו. ייתכן ויהיו לך כמה כתובות IP שהן חלק פיסי מאותה הרשת המחוברת, ואילו כתובות אחרות יהיו חיבורים נפרדים מאתרים אחרים.

**22.** לאחר שסיימת, לחץ על לחצן Continue.

**23.** תוכנית ההתקנה של Windows NT 4.0 Option Pack תסיים את הגדרת MSMQ, וכל שיהיה עליך לעשות הוא ללחוץ על לחצן Finish.

## טיפ:

התקנת סוגי השרתים האחרים של MSMQ דומה להתקנת PEC.

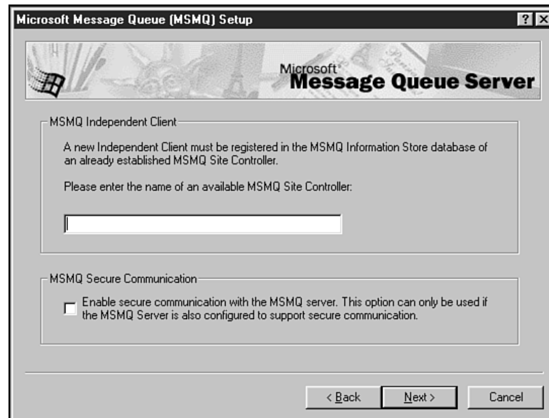
## התקנת לקוחות MSMQ

השלבים שבהתקנת לקוח MSMQ אינם דורשים שיתוף פעולה רב מצד המתקין כמו שבהתקנת שרת MSMQ. זכור כי ישנם שני סוגי לקוחות אותם אתה יכול להתקין (עצמאי ותלוי). חוץ מזה, תהליך התקנת שני סוגי הלקוחות זהה:

1. פתח את תוכנית ההתקנה של Windows NT 4.0 Option Pack ובחר באפשרות Add/Remove Components.
2. אתר את תיבת הסימון Microsoft Message Queue בתיבת הרשימה Components וסמן אותה.
3. לחץ על לחצן Show Subcomponents כדי לפתוח את תיבת הדו-שיח Microsoft Message Queue, המוצגת בתרשים 22.7.
4. סמן את כל ארבעת תיבות הסימון: HTML Documentation ,Administration Tools, Microsoft Message Queue Core ו-Software Development Kit.
5. לחץ OK כדי להחיל את השינויים שביצעת וכדי לחזור אל תיבת רשימת הרכיבים.
6. לחץ OK כדי לחזור אל תיבת הדו-שיח Windows NT 4.0 Option Pack.
7. לחץ על לחצן Next כדי להתקדם למסך הבא, המוצג בתרשים 22.15.

**תרשים 22.15** באפשרותך לבחור בין שני סוגי לקוחות MSMQ.

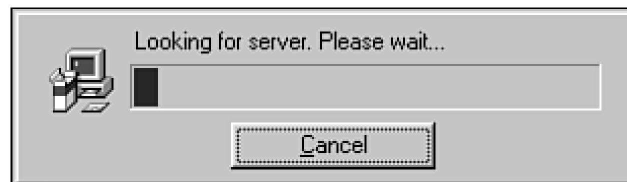
8. בחר בסוג הלקוח אותו אתה רוצה להתקין: Independet Client או Dependent Client.
9. לחץ על לחצן Next כדי להתקדם למסך דומה לזה שמוצג בתרשים 22.16.




**תרשים 22.16** במקרה זה, אתה מתקין לקוח עצמאי: עם זאת, המסכים זהים למסכי התקנת לקוח תלוי.

**10.** הקלד את שם מחשב בקר האתר של MSMQ שלך בלוח העליון. זהו השרת אותו התקנת בסעיף הקודם.

**11.** לחץ על לחצן Next כדי להורות לתוכנית ההתקנה של MSMQ לאתר את השרת, כפי שמוצג בתרשים 22.17.




**תרשים 22.17** בזמן שהמערכת שלך מנסה ליצור קשר עם בקר האתר, מוצג מחוון התקדמות.

**12.** לאחר שהשרת נמצא, תעבור למסך שמוצג בתרשים 22.10 אשר מציין את תיקיית ההתקנה של קבצי MSMQ. הקלד את נתיב התיקיה עבור קבצי MSMQ ולחץ על לחצן Next כדי להתחיל את ההתקנה.

**13.** לאחר שתהליך ההתקנה הסתיים, לחץ על לחצן Finish כדי לסגור את תוכנית ההתקנה Windows NT 4.0 Option Pack.

## הערה:

זכור כי אתה יכול לבחור בלקוח תלוי (Dependent) רק אם אתה בטוח שהלקוח יוכל תמיד להשיג את השרת, או שההודעות שהלקוח מוסר ומקבל אינן חיוניות בדרגה קריטית לפעולת היישום שלך.

# שימוש בדפי ASP עם MSMQ

כשאתה כבר מצויד בידע על עקרונות פעולת MSMQ, אתה יכול להתחיל בפיתוח MSMQ. שילוב MSMQ ביישומי האינטרנט שלך דורש יותר תכנון מאשר קידוד ממשי. תכנון קפדני הוא המפתח להצלחה ב-MSMQ. כדי להתוודע ל-MSMQ ו-ASP, תיצור יישום אינטרנט לדוגמה שעוסק בעיבוד הזמנות. בדוגמה זו, תשתמש בארכיטקטורה מפושטת המבוססת על מערכת אחת שמתפקדת הן כשרת MSMQ והן כלקוח. אם יש לך מחשב זמין נוסף, תוכל להרחיב את ההוראות כך שהדוגמה תשתמש בשני המחשבים. בדוגמה זו, אנו מניחים ששרת MSMQ (וכן בקר PEC) הותקן.

**הערה:**

שרת MSMQ יכול לתפקד גם כלקוח MSMQ.

## הבנת רכיבי MSMQ

בלב התקשורות של MSMQ נמצאים רכיבי MSMQ. רכיבים אלה דומים למדי לכל רכיב אחר בו אתה משתמש בדפי ה-ASP שלך או לחילופין בתוכנית הנוצרת בשפה אחרת, כגון Visual Basic. ישנם עשרה רכיבי MSMQ שונים שמותקנים, כשכל אחד מהם מספק גישה ל-MSMQ API (Application Programming Interface). בהמשך סעיף זה, תתנסה בכמה מרכיבים אלה בדוגמת ה-ASP שלך. להלן תיאור קצר של כל אחד מהרכיבים הללו:

- MSMQQuery - משמש ליצירת שאילתת MSMQ Information Store (MQIS) עבור כל תור הודעות קיים. תוצאות השאילתה מוחזרים דרך MSMQQuery בצורת אובייקט MSMQQueueInfos.
- MSMQQueueInfos - משמש לבחירת תור ציבורי מתוך אוסף של תורים המוחזרים על ידי MSMQQuery.
- MSMQQueueInfo - מיועד לניהול תורים, יצירת ומחיקת תורים, ושינוי מאפייניהם.
- MSMQQueue - מיועד לסיור בתוכן תור ההודעות, בצורה דומה מאוד לערכת רשומות (Recordset) של בסיסי נתונים.

- MSMQEvent - מיועד להטמעת אירועים בתורים עבור פעולות תורים, כגון הגעת הודעה, שגיאות, ואירועים אחרים.
- MSMQMessages - מיועד ליצירת הודעות MSMQ עבור תור הודעות.
- MSMQCoordinatedTransactionDispenser - מיועד ליצירת אובייקט DTC tranceation, המוחזר כאובייקט MSMQTransaction.
- MSMQTransaction - מיועד לחיוב או ביטול תנועות בתוך תור הודעות, תוך התבססות על אובייקט DTC transaction או תנועה פנימית.
- MSMQTransactionDispenser - מיועד ליצירת אובייקט תנועה (שלא מסוג DTC) בו ניתן להשתמש לצורך משלוח וקבלת הודעות.
- MSMQApplication - מיועד לקביעת זהות מחשב מסוים (לקוח או שרת).

**הערה:**

לקבלת התייחסות לכל רכיב מזווית התכנות, פנה לתיעוד Windows NT 4.0 Option Pack.

# שימוש ב-MSMQ בדפי ASP

כעת, לאחר שהתקנת שרת ולקוח MSMQ, אתה יכול להתחיל להשתמש ב-MSMQ בתוך יישומי האינטרנט שלך. בסעיף זה, תיצור דוגמת MSMQ בסיסית על ידי ASP, תוך שימוש ברכיבי MSMQ. דוגמה זו מתוכננת בסביבת פיתוח בה הלקוח והשרת שוכנים באותו המחשב.

**הערה:**

אם אתה רוצה להשיג מעבר לכך, תוכל להתאים דוגמה זו לעבודה עם שרת ולקוח נפרדים. עם זאת, אני אתמקד בעקרונות ואניח לך לעשות זאת בזמן אחר.

# תכנון התקשורת

ליבה של הדוגמה הוא מערכת פשוטה לעיבוד הזמנות של חברת תעופה, SpudAir. SpudAir היא חברת תעופה אזורית בעלת תקציב צנוע וזמן מוגבל להטמעת מערכת עיבוד הזמנות. התבקשת ליצור מערכת הזמנות פשוטה עבור אנשי המכירות של החברה כדי לאפשר להם לעבד דרישות לכרטיסי טיסה. תהליך עיבוד ההזמנות בחברת SpudAir פשוט מאוד :

1. נציגי המכירות של SpudAir מקבלים דרישה מלקוח לכרטיס בטיסה זמינה. נציגי המכירות סורקים את בסיס הנתונים המקוון המכיל את הטיסות העתידיות ומאתרים מושב זמין עבור הלקוח.
2. נציגי המכירות מזינים טופס מקוון במערכת עיבוד ההזמנות, ודורשים שהמושב יישמר עבור הלקוח במערכת ההזמנות הפנימית של SpudAir.
3. הדרישה נשלחת בצורה אלקטרונית למערכת ההזמנות, אשר מסמנת את המושב כשמור.
4. מערכת ההזמנות שומרת את המושב ושולחת לנציגי המכירות הודעה המציינת שהמושב נשמר.
5. נציגי המכירות מודיעים ללקוח שהמושב שלו נשמר ומבצעים תהליך חיוב לקוח בעלות הכרטיס.

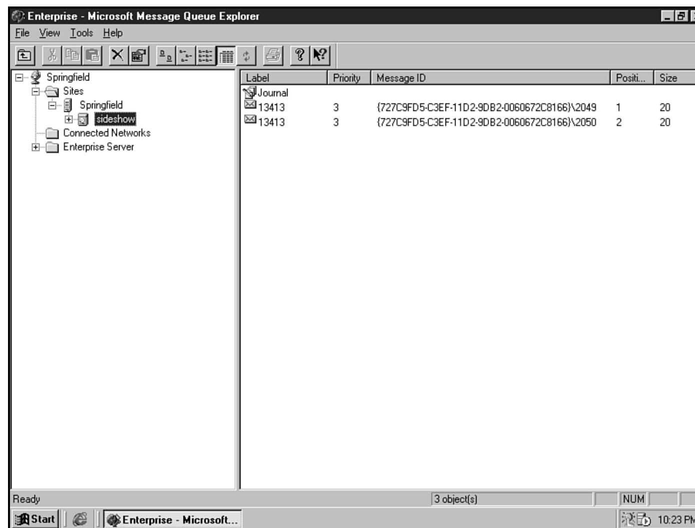
לצורך דוגמה זו, תיצור מודל פשוט של תהליך עיבוד זה.

# יצירת תור ההודעות

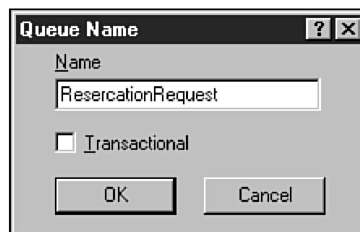
השלב הראשון ביצירת מערכת העיבוד שלך היא ליצור את תורי ההודעות בהם ישתמש יישום האינטרנט. אם תתבונן בתהליך העיבוד של נתוני חברת SpudAir, תוכל לזהות תור הודעות אותו ניתן ליצור, תור הדרישות להזמנות. לשם יצירת התור, בצע את השלבים הבאים:

1. פתח את Microsoft Message Queue Explorer. MSMQ Explorer מוצג בתרשים 22.18.
2. אתר את תיקיית Sites בלוח השמאלי. לחץ עליה כדי להרחיבה לאתר MSMQ אותו יצרת כשהתקנת את MSMQ.
3. הרחב את אתר MSMQ שלך כך שיציג את שרת MSMQ.
4. לחץ לחיצה ימנית על שרת MSMQ שלך. בחר באפשרות New, Queue מתוך תפריט התוכן. כעת תופיע תיבת הדו-שיח Queue Name, כפי שמוצג בתרשים 22.19.
5. הקלד את שם התור החדש אותו אתה רוצה ליצור. לדוגמה, השתמש בשם ReservationRequest.
6. לחץ OK כדי להחיל את השינויים שביצעת וכדי ליצור את תור ההודעות. התור מתווסף לרשימת התורים, כפי שמוצג בתרשים 22.20.

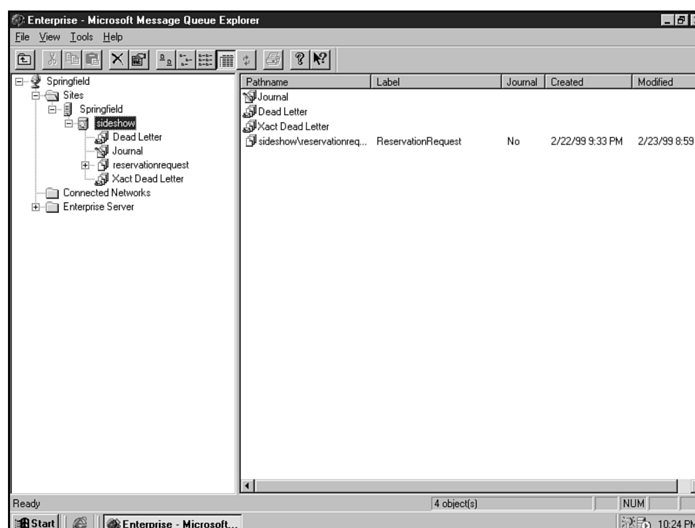
לאחר שיצרת את תורי ההודעות הדרושים לך, אתה יכול להמשיך למשלוח הודעות מדפי ה-ASP.




**תרשים 22.18** Microsoft Message Queue Explorer הוא מסוף ניהול המערכת של MSMQ.




**תרשים 22.19** תיבת הדו-שיח Queue Name מאפשרת לך ליצור תור תנועות.




**תרשים 22.20** כל תור מופיע ברשימה ההיררכית בדומה לתצוגת התיקיות בסייר Windows.

# משלוח הודעה

לפני שאתה יכול לשלוח הודעה לתור, יש ליצור קו תקשורת עם שרת MSMQ. פעולה זו נקראת בשם **פתיחת תור**. התחל ביצירת דף האינטרנט שיפתח את התור:

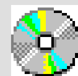
1. צור דף ASP חדש בעורך שלך וקרא לו בשם msmq.asp.
2. העתק את תוכן תדפיס 22.1 אל דף msmq.asp אותו יצרת.

**תדפיס 22.1** דף msmq.asp

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<TITLE>MSMQ Example: Step 2</TITLE>
</HEAD>
<BODY BGCOLOR="White">
<FONT FACE="Arial,Helvetica,Sans-Serif">
<H1>Open an MSMQ Public Queue</H1>
<P ALIGN="LEFT">
This page creates the connection to the MSQS message queue for all
activity. </p>
<HR SIZE=1 NOSHADE>
<%
set objMSMQ = Server.CreateObject ("MSMQ.MSMQQueueInfo")
objMSMQ.PathName = ".\ReservationRequest"
objMSMQ.Label = "ReservationRequest"
On Error Resume Next
objMSMQ.Create
On Error Goto 0
set objMsgQueue = objMSMQ.Open (2, 0)
%>
<HR SIZE=1 NOSHADE>
</FONT>
</BODY>
</HTML>
```

3. שמור את השינויים בקובץ msmq.asp.

**הערה:**



קובץ msmq.asp נמצא בתקליטור המצורף.

קוד קצר זה מתחיל את עבודתך עם אובייקטי MSMQ. בשורה 14, אתה יוצר מופע של האובייקט MSMQ.MSMQQueueInfo המאפשר לך לפתוח את תור ReservationRequest. שורות 15 ו-16 מתייחסות לתור עצמו. שים לב למאפיין objMSMQ.PathName בשורה 15, היכן שההתייחסות לתור דומה למדי להתייחסות לתיקיה פיסית במערכת הקבצים שלך. בשורות 17-19, אתה יוצר את תור ההודעה בצורת אל-כשל (Fail-Safe). למעשה, יצרת את התור בסדרת השלבים האחרונה. עם זאת, תוכל ליצור אותו מחדש כדי להבטיח שהוא קיים. אם התור קיים, כפי שהוא אמור להיות, פקודה זו לא תבוצע. לבסוף, בשורה 20, אתה פותח ממשית את התור למשלוח הודעה. כפי שתוכל לראות בשורה 20, מועברים לתור שני פרמטרים : Access ו-ShareMode. הערכים הללו מוצגים בטבלה 22.1 ו-22.2.

**טבלה 22.1** ערכי הפרמטרים Open Access

| קבוע | ערך | תיאור |
|---|---|---|
| MQ_PEEK_ACCESS | 0 | גישת קריאה-בלבד אל התור |
| MQ_SEND_ACCESS | 1 | גישת כתיבה בלבד אל התור |
| MQ_RECEIVE_ACCESS | 2 | גישת קריאה\כתיבה אל התור |

**טבלה 22.2** ערכי הפרמטרים של שיטת Open ShareMode

| קבוע | ערך | תיאור |
|---|---|---|
| MQ_DENY_NONE | 0 | התור זמין לכולם |
| MQ_DENY_RECEIVE_SHARE | 1 | הגישה אל התור מוגבלת |

**הערה:**

למידע נוסף אודות הפרמטרים, פנה לתיעוד Windows NT 4.0 Option Pack.

פתח את הדף בדפדפן אינטרנט, ותיווכח שאין כל כך במה להביט. ההמשך יגיע מיד.

השלב הבא הוא ליצור ולשלוח את ההודעה עצמה. לשם כך, תסתמך על טופס שידחף את תוכן ההודעה אל הדף msmq.asp שיצרת. בצע את השלבים הבאים :

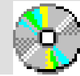
1. צור דף חדש בעורך שלך וקרא לו בשם msmq-form.htm.
2. הכנס את תדפיס 22.2 לדף החדש שיצרת.

**תדפיס 22.2** דף msmq-form.htm.

```
<HTML>
<HEAD>
<TITLE>MSMQ Form</TITLE>
</HEAD>
<BODY>
<FONT FACE="Arial">
<H1>MSMQ Example: Step 1</H1>
<HR SIZE=1 NOSHADE>
<P ALIGN="LEFT">
This form will be used to submit a message to the ReservationRequest
message queue.
</P>
<HR SIZE=1 NOSHADE>
<FORM action="msmq.asp" method=POST id=frmData name=frmData>
<TABLE BORDER="0" ALIGN="CENTER">
<TR>
<TD>
Customer Seating Position
</TD>
<TD>
<INPUT type="text" id=txtLabel name=txtLabel size=5 maxlength=5>
</TD>
</TR>
<TR>
<TD>
Customer Name
</TD>
<TD>
<INPUT type="text" id=txtBody name=txtBody size=50 maxlength=50>
</TD>
</TR>
<TR>
<TD COLSPAN=2 ALIGN="CENTER">
<INPUT type="submit" value="Submit" id=btnSubmit name=btnSubmit>
</TD>
</TR>
</TABLE>
</FORM>
<HR SIZE=1 NOSHADE>
<P ALIGN="LEFT">
This page pushes information to the <U>msmq.asp</U> page for
processing.
</P>
</FONT>
</BODY>
</HTML>
```

**3.** שמור את השינויים שביצעת בקובץ msmq-form.htm.

**הערה:**


הקובץ msmq-form.htm נמצא בתקליטור המצורף.

דף פשוט זה ידחף את תוכן הטופס אל דף msmq.asp, כפי שמוצג בשורה 14. הטופס מורכב משתי תיבות טקסט, txtLabel ו-txtBody.

השלב הבא הוא שינוי דף msmq.asp שיצרת כך שיקבל את גוף ההודעה והתווית מדף msmq-form.htm וישלח את ההודעה אל תור ההודעה של MSMQ. לשם כך, בצע את השלבים הבאים:

1. פתח את דף msmq.asp בעורך שלך כדי לבצע את השינויים.
2. אתר את תחילת בלוק תסריט ASP בשורה 10 ואת סוף הבלוק בשורה 18.
3. בחר בבלוק התסריט כולו והחלף אותו בתדפיס 22.3.

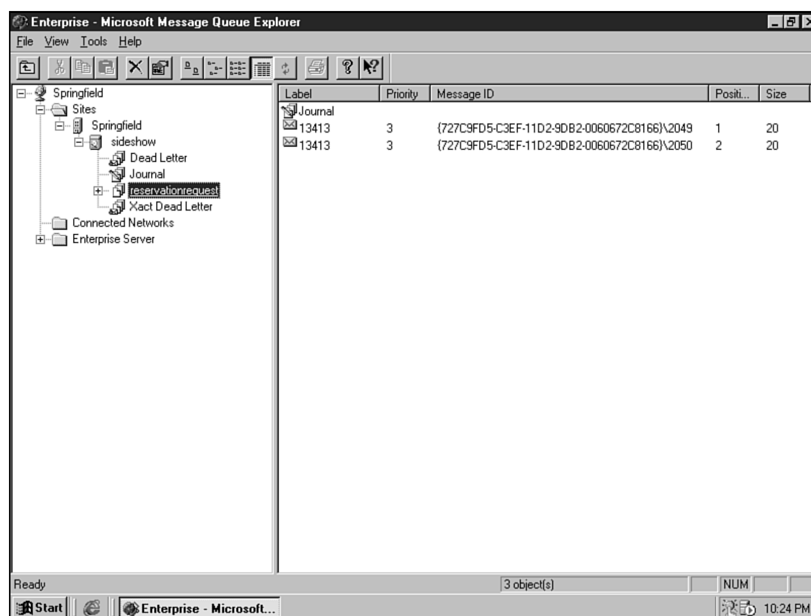
**תדפיס 22.3** בלוק התסריט msmq.asp המעודכן.

```
1   <%
2   On Error Resume Next
3   Dim objMSMQ, objMsgQueue, objMessage, strCaption, strBody
4   strLabel = Request.Form("txtLabel")
5   strBody = Request.Form("txtBody")
6   set objMSMQ = Server.CreateObject ("MSMQ.MSMQQueueInfo")
7   objMSMQ.PathName = ".\ReservationRequest"
8   objMSMQ.Label = "ReservationRequest"
9   objMSMQ.Create
10  On Error Goto 0
11  set objMsgQueue = objMSMQ.Open (2, 0)
12  If objMsgQueue.IsOpen Then
13  Set objMessage = Server.CreateObject("MSMQ.MSMQMessage")
14  objMessage.Body = strBody
15  objMessage.Label = strLabel
16  objMessage.Send objMsgQueue
17  End If
18  objMsgQueue.Close
19  %>
```

4. שמור את השינויים שביצעת בדף msmq.asp.

בשורות 4-5, ערכי תיבת הטקסט המועברים מדף msmq-form.htm מאוחסנים במשתני מחרוזת תוך שימוש באובייקט Request. לאחר שגוף ההודעה והתווית שלה הוקצו למשתנים, ההודעה מוכנה למשלוח. התהליך מתחיל בשורה 13 ביצירת אובייקט MSMQ.MSMQMEssage בשם objMessage. שורות 14-15 מגדירות את ההודעה על ידי הקצאת הגוף והתווית בהתאמה. כדי לשלוח את ההודעה, שורה 16 קוראת לפונקציה ObjMessage.Send. לבסוף, בשורה 18, החיבור (connection) לתור ההודעות נסגר.

לאחר שמסרת כמה הודעות, פתח את MSMQ Explorer ולחץ על תור Reservationrequest. כעת תבחין בכל אחת מההודעות המופיעות בלוח הימני, המזוהות על ידי מספר ייחודי, כפי שמוצג בתרשים 22.21.




**תרשים 22.21** ניתן לפתוח כל הודעה בתור תוך שימוש ב-MSMQ Explorer.

# קבלת הודעה

החלק האחרון במבוך MSMQ הוא קבלת ההודעה. כדי לבצע את השלב הזה, תיצור דף שלישי שיקבל ויציג את תוכן תור ההודעות. MSMQ מציע שלוש דרכים לקריאת הודעות מתור ההודעות: סינכרונית, אסינכרונית, ועל ידי שימוש בסמן (בדומה לבסיס נתונים). עם זאת, ASP תומך רק בשיטה הסינכרונית, ולכן הפעלת התסריט שלך תיחסם עד שתהיה הודעה זמינה או עד שהתסריט שלך ייצור שגיאת Time Out בזמן ניסיון התחברות לשרת MSMQ.

כדי ליצור את הדף החדש שיקרא את תוכן תור ההודעות, בצע את השלבים הבאים:

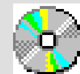
1. צור דף חדש בעורך שלך וקרא לו בשם msmq-read.asp.
2. הכנס את תדפיס 22.4 אל תוך הדף החדש שיצרת.

**תדפיס 22.4** דף msmq-read.asp.

```
1   <HTML>
2   <HEAD>
3   <TITLE>MSMQ Read</TITLE>
4   </HEAD>
5   <BODY>
6   <FONT FACE="Arial">
7   <H1>MSMQ Example: Step 3</H1>
8   <HR SIZE=1 NOSHADE>
9   <P ALIGN="LEFT">
10  This page reads the waiting messages in the ReservationRequest
11  message queue and returns it to the screen.
12  </P>
13  <HR SIZE=1 NOSHADE>
14  <%
15  set objMSMQ = Server.CreateObject("MSMQ.MSMQQueueInfo")
16  objMSMQ.PathName = ".\ReservationRequest"
17  set objMessageQueue = objMSMQ.Open(1,0)
18  Do While True
19  Set objMessage = objMessageQueue.Receive(false, true, 1000)
20  If objMessage Is Nothing Then Exit Do
21  Response.Write("Customer: " & objMessage.Body & "<BR>")
22  Response.Write("Seat: " & objMessage.Label & "<BR> <BR>")
23  Loop
24  %>
25  <HR SIZE=1 NOSHADE>
26  <P ALIGN="LEFT">
27  This page pulls the message information from the message queue.
28  </P>
29  </FONT>
30  </BODY>
31  </HTML>
```

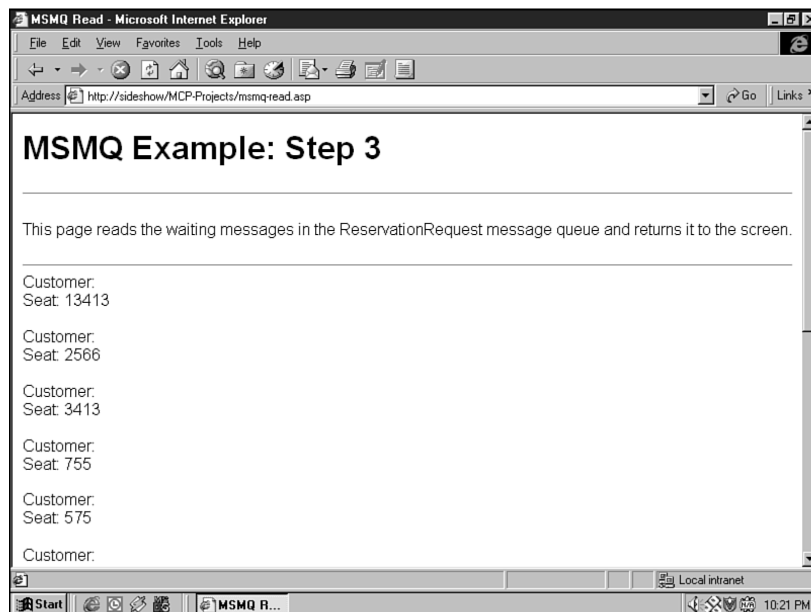
**.3** שמור את השינויים שביצעת בקובץ msmq-read.asp.

**הערה:**



הדף msmq-read.asp נמצא בתקליטור המצורף.

בשורות 15-17, אתה יוצר חיבור (connection) לתור ההודעות Reservationrequest. לאחר מכן, אתה יוצר את האובייקט objMessage אשר מייצג את ההודעה עצמה בשורה 19, ובהנחה שתוכן ההודעה אינו ריק, אתה כותב את תוכן גוף ותווית ההודעה על גבי המסך. דבר זה מבוצע על ידי לולאת ...Do-While המציגה את כל ההודעה הממתינות בתור, כפי שמוצג בתרשים 22.22.




**תרשים 22.22** כל הודעה בתור מוצגת לפני המבקר.

אם אין הודעות בתור (לדוגמה, אם לאחר צפייה בדף אתה מרענן אותו לפני שהצבת עוד הודעות בתור), הדף ייתקע. דבר זה נובע מצורת העברת ההודעות הסינכרונית מתור ההודעות של MSMQ, הממתין להודעה זמינה או לשגיאת Time Out בחיבור לשרת. יישום אינטרנט מעוצב היטב אמור להשתמש ב-Microsoft Transaction Server כדי להבטיח שהתקשורת תתקיים. במידה ולא ניתן לקיים תקשורת, עליו להבטיח שהמשתמש לא יוטרד על ידי השהיות או הודעות שגיאה.

# סיכום

הדוגמה הפשוטה בפרק זה מדגימה כמה קל לשלב את MSMQ אל תוך יישומי האינטרנט שלך. על ידי שימוש ב-MSMQ לניהול תקשורות העיבודים שלך, יישומי האינטרנט משיגים רמת עצמאות חדשה ו-fault tolerance. בכך גם מופחתת המורכבות שבקידוד מערכת הודעות משלך. MSMQ מתאים במיוחד לסביבת רשת מבוזרת ולמספר מערכות המשתפות ביניהן נתונים או דרישות. דוגמה אידיאלית לכך היא עולם הבנקאות, בו הודעות פיננסיות ודרישות חייבות לעבור בשלמותן או לא לעבור כלל. על ידי שימוש בעוצמת שירותי תורי ההודעות, המוסדות הפיננסיים יכולים להסתמך על הארכיטקטורה שמאחורי מערכת ההודעות כדי לטפל בפרטים הקטנים של העברת הנתונים. דבר זה מאפשר למפתחים להתמקד במשימה הספציפית: ביצוע פעולות על הנתונים לאחר קבלתם.

# חלק 6

# נושאים מתקדמים

**בחלק זה:**

- **אבטחת אתר אינטרנט**
- **תחזוקת אתרי ASP**
- **שיפור ביצועי דפי ASP**
- **Win 2000, IIS 5.0, ASP - כולם ביחד**

## פרק 23

# אבטחת אתר אינטרנט

**בפרק זה:**

- **הצפנה, אימות ושלמות הנתונים עם SSL**
- **הגנת אתר אינטרנט באמצעות סיסמה**

כל מפתח אתרי אינטרנט חייב לדאוג לגבי בעיות באבטחת האתר. אם האתר שלך מארח מידע חסוי, כגון מספרי כרטיסי אשראי, מסמכים עסקיים או סיסמאות, מוטלת עליך האחריות להגן על מידע זה. כישלון באבטחה יכול לגרור תוצאות הרות אסון, גם לעסק שלך וגם למשתמשים באתר.

פרק זה מתמקד בשתי בעיות אבטחה. בחלקו הראשון של הפרק, תלמד כיצד להגן על אבטחת הנתונים בזמן העברתם ברשת האינטרנט. אם משתמש מקליד סיסמה או מספר כרטיס אשראי עליך למנוע ממי שאינו מורשה לכך מלגשת למידע זה בעודו עובר באינטרנט, או ברשת המקומית. חלקו הראשון של הפרק יסביר כיצד להשתמש ב-**SSL** **Secure Sockets Layer** כדי לספק ערוץ תקשורת מאובטח, וכן יכסה שיטות שונות לשימוש ב-SSL בדפי ASP.

**הערה:**

בשרת IIS נחשפו מספר חורי אבטחה משמעותיים. לדוגמה, נמצא שכל משתמש יכול לראות את קוד המקור של דף ASP על ידי הוספת נקודה בסיום כתובת הדף. ניתן להשתמש בשיטה זו כדי לראות מידע חסוי, כגון מסד נתוני סיסמאות. באתר הבית שלה מספקת Microsoft טלאים (Patches) לתיקון בעיות אלו. כדי לראות את עדכוני האבטחה והטלאים האחרונים גלוש לאתר:
http://www.microsoft.com/security

חלקו הנוסף של הפרק מתמקד בבעיית הגנה על אתר האינטרנט שלך באמצעות סיסמה. נידונות שתי גישות שונות לבעיה זו. תלמד כיצד להגן בסיסמה על דפי ה-ASP שלך, תוך שימוש במסד נתוני אבטחה ובאמצעות אבטחת Windows NT.

# הצפנה, אימות ושלמות הנתונים עם SSL

**Secure Sockets Layer** (SSL) הוא פרוטוקול אשר פותח במקור על ידי Netscape לשם העברה מאובטחת של נתונים ברשת שאינה מאובטחת. SSL הוא השיטה היחידה הקיימת לשליחת מידע פרטי ברשת האינטרנט, ואשר פועלת עם רוב הדפדפנים העדכניים. SSL מספק פתרון טכני לשלוש בעיות אבטחה נפרדות: הצפנה, אימות ושלמות הנתונים.

**הערה:**

שרת IIS של Microsoft ו- Internet Explorer (גירסה 3.0 ומעלה) תומכים גם בפרוטוקול Private Communication Technology (PCT, טכנולוגיית תקשורת פרטית). Microsoft פיתחה את PCT כגירסה יעילה יותר של SSL.

## Encryption

כאשר אתה ממלא בטופס HTML נתונים ושולח אותו לאתר אינטרנט, נשלחים הנתונים מהדפדפן שלך לשרת שבאתר. כאשר הנתונים עוברים ברשת האינטרנט הם, בדרך כלל, עוברים דרך מספר צמתי ביניים. באופן תיאורטי, ניתן לקלוט ולקרוא את הנתונים שמולאו בטופס.

הבעיה דומה למצב בו גנרל גדול בצבא צריך לשלוח דרך שטחי האויב הודעה המכילה תוכניות סודיות. כאשר השליח עובר בשטח לא מוכר הוא עלול ליפול בשבי, והאויב יכול לגנוב ולקרוא את המידע המסווג.

הפתרון המתאים גם עבור הגנרל, וגם עבור משתמש הממלא נתונים בטופס HTML, הוא להצפין את המידע לפני שהוא נשלח דרך השטח העוין. אפילו אם ההודעה תילכד, פרטיות המידע מוגנת - אלא אם כן כמובן, נפרץ הקוד הסודי.

SSL מצפין מידע העובר בין שרת אינטרנט ודפדפן אינטרנט. המידע מוצפן באמצעות אלגוריתם הצפנה ציבורי ידוע ומפתח הצפנת session סודי. מספר הסיביות במפתח ה-session קובע את עוצמת ההצפנה.

כאשר אתה מתקין IIS, אתה מתקין כברירת מחדל גרסת IIS התומכת במפתח הצפנת session באורך 40bits. בכל אופן, באפשרותך לשדרג את IIS כדי לתמוך במפתח חזק יותר באורך 128bits. למרות שהודעות שהוצפנו באמצעות מפתח של 40bits נפרצו כבר, הודעות המוצפנות במפתח באורך 128bits נחשבות כבלתי-פריצות בטכנולוגיה העכשווית.

מדוע, אם כן, לא להשתמש תמיד במפתח באורך 128bits? יש לכך שתי סיבות. הראשונה, תקשורת המשתמשת במפתח באורך 128bits יכולה להיות איטית משמעותית מזו המשתמשת במפתח באורך 40bits. ככל שהמפתח ארוך יותר, כך צריכה להתבצע עבודה רבה יותר על ידי השרת והדפדפן כדי להצפין ולפענח את ההודעה.

כמו כן, עד לפני זמן קצר היה השימוש במפתחות בני 128bits מוגבל לתחומי ארה״ב וצפון אמריקה בלבד. הממשל האמריקאי סיווג הצפנה מסוג זה כרגישה מבחינה אסטרטגית, וההתייחסות אליה היתה כאל כלי נשק צבאי לכל דבר. משמעות הדבר היתה שאין זה חוקי, למעט חריגים בודדים, לייצא תוכנית כלשהי התומכת בהצפנה ברמה זו אל מחוץ לתחומי ארה״ב (היתה גם גירסה מיוחדת עבור תושבי קנדה). חוק זה כלל שרתים ודפדפנים כאחד. רק לאחרונה איפשר הממשל האמריקאי לתושבי מדינת ישראל (ומספר ארצות אחרות) להוריד, להתקין ולהשתמש בהצפנה זו. למרות

זאת, עד לכתיבת שורות אלו עדיין קיימות חברות היוצרות/מפיצות תוכנות הצפנה שאינן מאפשרות למי שאינו תושב ארה"ב להוריד את המוצר שלהם.

בדרך כלל, אם אתה מתקין מפתח session באורך 128bits בשרת האינטרנט שלך, השרת יישא וייתן בצורה אוטומטית על רמת ההצפנה הגבוהה ביותר לצורך אבטחת התקשורת. אם מישהו מתקשר באמצעות דפדפן ובו מפתח באורך 40bits, השרת שלך ישתמש ברמת הצפנה זו באופן אוטומטי. בכל אופן, ניתן להגדיר את IIS כך שידחה דפדפנים שאינם תומכים במפתח באורך 128bits.

**הערה:**

כדי להתקין את המפתח החזק יותר של 128bits של SSL, תצטרך להשיג את Microsoft Encryption Pack. לאחר התקנת ההצפנה החזקה יותר לא תוכל להמשיך להתקין את Windows NT Service Pack הרגיל. למידע נוסף ראה מאמר שמספרו Q159709 ב-Microsoft Knowledge Base.

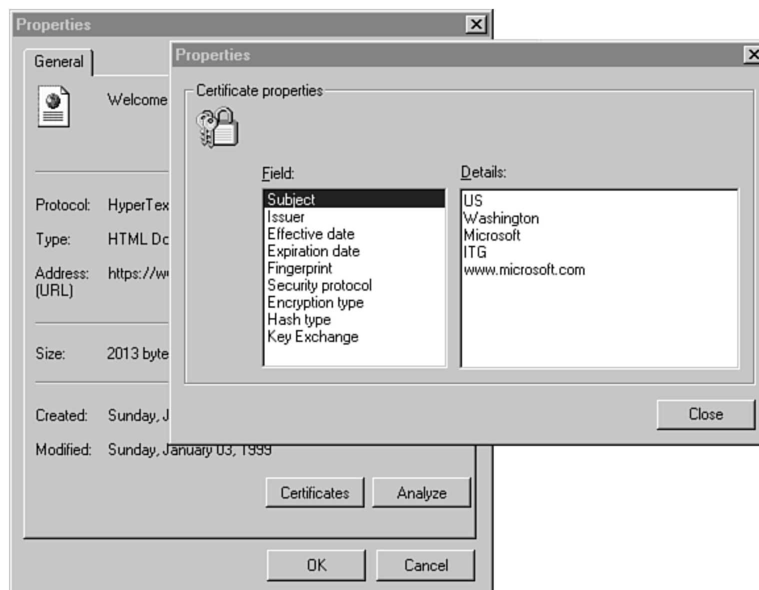
# Authentication

אם אתה מבקר באתר אינטרנט של הוצאת הוד-עמי בכתובת www.hod-ami.co.il אתה יכול להיות בטוח בנתינת פרטי כרטיס האשראי שלך כדי לרכוש ספר. אולם, גנב מתוחכם יכול ליצור אתר אינטרנט שלא ניתן יהיה להבדיל בינו לבין האתר המקורי, למשל של Amazon.com, ולגנוב את נתוני כרטיס האשראי שלך.

בחזרה לדוגמה הקודמת, זו של הגנרל שצריך לשלוח תוכניות סודיות דרך קווי האויב, תאר לך שהאויב החליט להתחזות ללקוח המיועד של התוכניות הסודיות. הגנרל והמתחזה החליטו על קוד סודי, והשליח מעביר את ההודעה המוצפנת באמצעות הקוד. אולם, השליח העביר את התוכניות הסודיות היישר למתחזה.

כדי למנוע מאתר אינטרנט להתחזות לאתר אחר, יכול SSL לשמש ל**אימות** (Authentication) אתר האינטרנט. כאשר אתה מתקין את SSL על שרת האינטרנט, אתה חייב להתקין תעודת **אישור שרת** (Server Certificate). אישור זה משמש לאימות זהות האתר, בדומה לאופן בו רשיון הנהיגה או הדרכון שלך משמשים לאימות זהותך האישית. אישור שרת מכיל מידע אודות הארגון, אתר האינטרנט והרשות המנפיקה את האישור שלך.

כדי לעבוד כמו **חתימה דיגיטלית** (Digital ID), חייב אישור השרת להיות מוחתם על ידי רשות אישורים. **רשות אישורים** (Certificate Authority) פועלת כצד שלישי אמין, המאמת את זהות אתר האינטרנט עבור המשתמשים בשירותיו. בכל פעם שאתה פותח דף באמצעות SSL, נכלל המידע מאישור השרת. לדוגמה, תוכל לראות את מידע האישור של דף הבית של אתר Microsoft בעת הצפייה בו ב-Internet Explorer. בסרגל הכתובת של הדפדפן הקלד את הכתובת https://www.microsoft.com. כאשר האתר מוצג בחלון הדפדפן פתח את תפריט File, בחר Properties ולחץ על Certificates (ראה תרשים 23.1).




**תרשים 23.1** אישור שרת עבור האתר של **Microsoft**.

במקום להשתמש ב**רשות אישורים** (Certification Authority, הידוע בקיצור כ-CA) של צד שלישי, תוכל גם להנפיק ולחתום על אישור משלך, תוך שימוש ב-Microsoft Certificate Server. במילים אחרות, באפשרותך להיות רשות האישורים של עצמך. אפשרות זו היא בעלת ערך רב כאשר אתה צריך לאמת מחשבים רבים בארגון שלך לחברים בארגון. בכל אופן, אם אתר האינטרנט שלך הוא ציבורי, עליך להשתמש ברשות אישורים של צד שלישי, כדוגמת VeriSign, מפני שאישור שרת הוא דבר אמין כמו הגוף המנפיק אותו.

SSL גירסה 3.0 תומך גם באישורי לקוח. **אישורי לקוח** (Client Certificates) עובדים בדיוק באותה צורה כמו אישור שרת, חוץ מהעובדה שהם משמשים כדי לאמת דפדפני אינטרנט, ולא שרתי אינטרנט. גם Internet Explorer (גירסה 3.0 ומעלה) וגם Netscape Navigator (גירסה 3.0 ומעלה) תומכים באישורי לקוח. ניתן לקבל אישור לקוח מרשות אישורים (CA), או להשתמש ב-Microsoft Certificate Server כדי להנפיק אישורים משלך.

# שלמות נתונים

דמיין לעצמך שטיפוס זדוני כלשהו החליט לשנות הודעה בעודה מועברת ברשת האינטרנט. טיפוס זה אינו קורא את ההודעה ואף אינו מונע את העברתה. ההודעה פשוט מושחתת.

אם נחזור שוב לדוגמת הגנרל, הבה ונניח כי השליח הצליח להעביר את התוכניות הסודיות ללקוח המיועד. אולם, כאשר הוא עבר בשטח האויב הוחלפה ההודעה, ללא ידיעתו של השליח. התוצאה: התוכניות הלא נכונות הועברו ליעד.

SSL מגן על שלמות הנתונים בעודם מועברים באינטרנט. כאשר הודעות נשלחות באמצעות SSL, הן כוללות קוד אימות הודעה - Massage Authentication Code (MAC). קוד זה מאפשר לזהות אם הודעה שונתה. במילים אחרות, כאשר אתה משתמש ב-SSL אתה יכול להיות בטוח שההודעה שהתקבלה זהה להודעה שנשלחה.

## כמה מוגן הוא SSL?

כמה בטוח, אם כן, הוא פרוטוקול SSL? האם באפשרותך להשתמש בבטחה ב-SSL כדי להעביר ברשת האינטרנט מידע אודות כרטיס אשראי או מסמכים עסקיים/פרטיים? כל אתרי המסחר העיקריים באינטרנט המקבלים מידע לגבי כרטיסי אשראי משתמשים בפרוטוקול SSL. לדוגמה, Amazon.com מקבל מידע לגבי למעלה מ- 4.5 מליון כרטיסי אשראי של משתמשים, תוך שימוש ב-SSL.

התשובה האמיתית היא שאין לך הרבה ברירה. אם ברצונך להעביר מידע פרטי באינטרנט, מבלי להכריח את המשתמשים להשתמש בתוכנות מיוחדות כגון Wallet (תוכנה לקניות מקוונות), רכיבי ActiveX, או יישומוני Java, עליך להשתמש ב-SSL. SSL מהווה את השיטה היחידה להעברה של מידע פרטי, הנתמך על ידי מרבית הדפדפנים.

## הגדרת השרת לשימוש ב-SSL

הגדרת השרת לשימוש ב-SSL הוא תהליך פשוט. אולם, אם אתה מתכנן להשתמש באישור שרת מרשות אישורים של צד שלישי, ראה הוזהרת - התהליך עלול להיות ארוך. תהליך קבלת אישור שרת יכול להיארך עד כשמונה שבועות, והדבר אינו זול במיוחד. VeriSign, לדוגמה, גובה $340.00 עבור אישור שרת, ועליך לשלם $249.00 נוספים כל שנה, לחידוש האישור.

שלושה שלבים עיקריים מעורבים בהתקנת SSL. קודם כל, עליך ליצור **קובץ בקשת אישור** (Certificate Request File) **וקובץ צמד מפתחות אבטחה** באמצעות שימוש ב-Microsoft Key Manager. בשלב הבא, עליך להגיש בקשה לרשות אישורים של צד שלישי עבור אישור שרת. הדבר נעשה על ידי אספקת קובץ בקשת האישור אותו יצרת. לבסוף, לאחר שקיבלת את האישור, עליך להתקין אותו באמצעות Microsoft Key Manager.

## יצירת קובץ בקשת אישור

כדי ליצור קובץ בקשת אישור, הנקרא גם **בקשת חתימת אישור** - **Certificate Signing Request** (CSR), פתח את Microsoft Key Manager על ידי הפעלת Internet Service Manager ולחיצה על הסמל Key Manager. בחלון Key Manager פתח את תפריט Key ובחר Create New Key. בחירה זו תפעיל אשף אשר ינחה אותך במהלך משימת יצירת קובץ בקשת האישור (ראה תרשים 23.2).

**תרשים 23.2** Microsoft Key Manager.

כדי ליצור את קובץ בקשת האישור, ספק את הנתונים הבאים:

- Request File - כאשר תשלים את האשף, יאוחסן בכונן הקשיח שלך קובץ בקשת האישור בשם זה.
- Key Name - ניתן להקליד כאן כל שם העולה על דעתך. שם זה משמש לזיהוי המפתח.
- Password - יהיה עליך להשתמש בסיסמה זו כאשר תתקין את אישור השרת החתום, לאחר שתקבל אותו מרשות האישורים.
- Key Size - כברירת מחדל, Key Size יהיה באורך 512bits. אם אתה משתמש בגירסה הביתית של IIS, Key Size יהיה באורך 1024bits. Key Size מתייחס לעוצמת אישור השרת, לא לעוצמת מפתח ה-session המשמש להצפנת הודעות.
- Organization - שם בעליו של שם ה-domain שלך. בדרך כלל, Organization הוא שם החברה שלך.
- Organization Unit - שם המחלקה או היחידה הארגונית.
- Common Name - שם ה-domain המוסמך המלא שלך. לדוגמה, www.aspsite.com. אין לכלול את הפרוטוקול (//:HTTP).
- Country/Region - קוד ISO בעל שני תווים עבור הארץ שלך. לדוגמה, US עבור United State, CA עבור Canada או IL עבור ישראל. האשף מספק קישור לרשימת קודי הארצות (לשם כך צריך קישור פתוח לאינטרנט).
- State/Province - שמה המלא של המדינה או המחוז שלך. לדוגמה, California.
- Locality - שם העיר שלך. לדוגמה, San Francisco.
- Your Name - שמך המלא.

- Email Address - כתובת הדואר האלקטרוני שלך.
- Phone Number - מספר הטלפון שלך.

לאחר שסיפקת נתונים אלה לאשף, יישמר קובץ בקשת אישור בכונן הקשיח שלך. סמל של מפתח שבור יופיע ב-Key Manager המציין כי נוצר קובץ בקשת אישור, אולם האישור עדיין לא הותקן. אם אתה סקרן, הנה דוגמה לקובץ בקשת אישור:

```
WebMaster: webmaster@aspsite.com
Phone: 5555555555
Server: Microsoft Key Manager for IIS Version 4.0

Common-name: www.aspsite.com
Organization Unit: aspsite
Organization: aspsite
Locality: Sam Francisco
State: California
Country: US

- - - - - BEGIN NEW CERTIFICATE REQUEST - - - - -
FDrtiJLK5kfKGgutKKOgJoLKo66KuIIirOikfKJJo7SEddRNRTYUYlgK6UitypeKKF45
KKGHSER0DIIOErtUEY56hUFFGXx32sKJFY6ffDrUDSY1UEAx/wlKSDD59RTpDkf
P3tyyr54ffIU+oi/gf56KFKkfKGgutFFGXx32sKJIOErtUEYlgK6UKKI7+diFTTR998d
IIOErtUEY56hUFyyr54ffFGXx32sKJIO6UitypeKKFGHSER0DutKKOgJo gf56IITUYR
R0DutKKOgJr54ff/tIIYU87Swte45/er+ios4kkLLKIXRRtr3OHFNhfNVKQQw87DGC
+Zh/qwertyuiPLMKNBJHI64HFU756310gw==
- - - - - END NEW CERTIFICATE REQUEST - - - - -
```

# הגשת בקשה לאישור שרת

לאחר שיצרת קובץ בקשת אישור, באפשרותך להגיש בקשה לאישור השרת מרשות אישורים (CA). לפניך רשימה של שלוש רשויות האישורים הפופולריות ביותר:

- VeriSign Inc. (http://www.verisign.com)
- GTE CyberTrust Solusions (http://www.cybertrust.gte.com)
- Thawte Consulting (http://thawte.com)

לדוגמה, כדי להגיש בקשה עבור אישור שרת של VeriSign, גלוש לכתובת http://www.verisign.com ובחר ב-Web Site Services. כעת עליך לבחור בסוג האבטחה המבוקש. יהיה עליך לספק ל-VeriSign מידע מזהה על הארגון שלך, כגון מספר DUNS של דן וברדסטריט (אם קיים), חוזה התאגיד או רשיון העסק שלך. לאחר שתספק מידע זה, תוכל להעביר את קובץ בקשת האישור באמצעות טופס מקוון. לאחר שהמידע שמסרת יאומת, תקבל הודעת דואר אלקטרוני המכילה הוראות כיצד להעביר אליך את אישור השרת החדש.

## התקנת אישור שרת

השלב האחרון בהכנת השרת לתמיכה ב-SSL היא ההתקנה המעשית של אישור השרת. כדי להתקין את אישור השרת, הפעל שנית את Microsoft Key Manager, פתח את תפריט Key ובחר Install Key Certificate. פתח את קובץ אישור השרת מהכונן הקשיח וספק את הסיסמה בה השתמשת ליצירת קובץ בקשת האישור. כעת, ציין את כתובת ה-IP ומספר ה-port בו ישתמש השרת עבור SSL. לכשתסיים, יופיע סמל של מפתח שלם ב-Microsoft Key Manager.

אישור שרת תקף רק לזמן קבוע מראש. במסגרת הימנית של Microsoft Key Manager תוכל לראות את התאריך המדויק בו יפקע תוקף אישור השרת שלך. כדי להמשיך להשתמש ב-SSL עליך לבקש אישור שרת חדש לפני תאריך תפוגה זה.

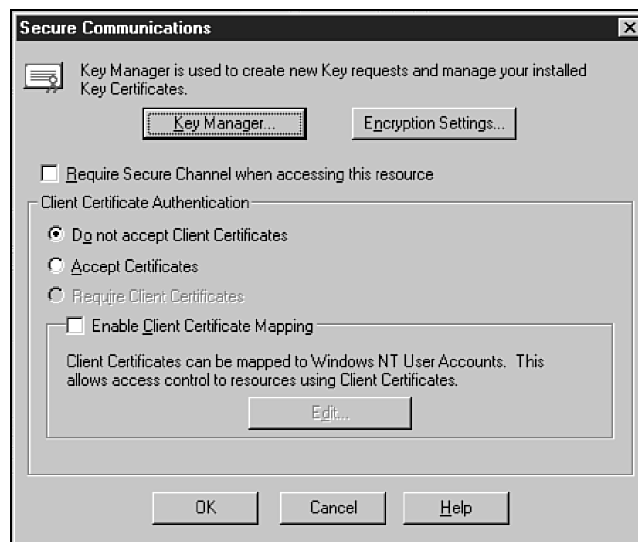
**הערה:**

אם אתה חייב להעביר את האישור שלך לשרת חדש, תוכל להשתמש ב- Microsoft Key Manager כדי ליצור עותק גיבוי של האישור. פתח את תפריט Key, בחר Export Key ובחר Backup File. כעת באפשרותך לטעון את האישור לשרת חדש על ידי בחירה ב- Key, Import Key, Backup File. לשרת החדש חייב להיות בדיוק אותו שם domain אינטרנט כמו לשרת המקורי (כתובת ה-IP יכולה להיות שונה).

## שימוש ב-SSL בדפי ASP

לאחר שהגדרת את השרת לעבודה עם SSL, באפשרותך לבקש כל דף מאתר האינטרנט שלך באופן מאובטח. כדי לאחזר דף אינטרנט באמצעות שימוש ב-SSL, השתמש בכתובת במבנה **https**://www.yourdomain.com/page.asp ולא באופן הרגיל **http**://www.yourdomain.com/page.asp. שיטה זאת תעבוד על כל דף באתר שלך.

ניתן לאלץ משתמשים להשתמש ב-SSL כאשר הם מבקשים דף מהאתר שלך. פתח את מאפייני (Properties) הספריה או דף בודד ב-Internet Service Manager ובחר בכרטיסיה Directory Services או בכרטיסיה File Security. כעת, לחץ על הלחצן Edit מתחת ל-Secure Communications ובחר ב-Require Secure Channel When Accessing This Resource (ראה תרשים 23.3). אם ברצונך לדרוש SSL ברמת הצפנה של 128bits, לחץ על Encryption Setting ובחר ב-Required 128-Bit Encryption.




**תרשים 23.3** דרישה ל-SSL כאשר מתבצעת בקשה לדף אינטרנט.

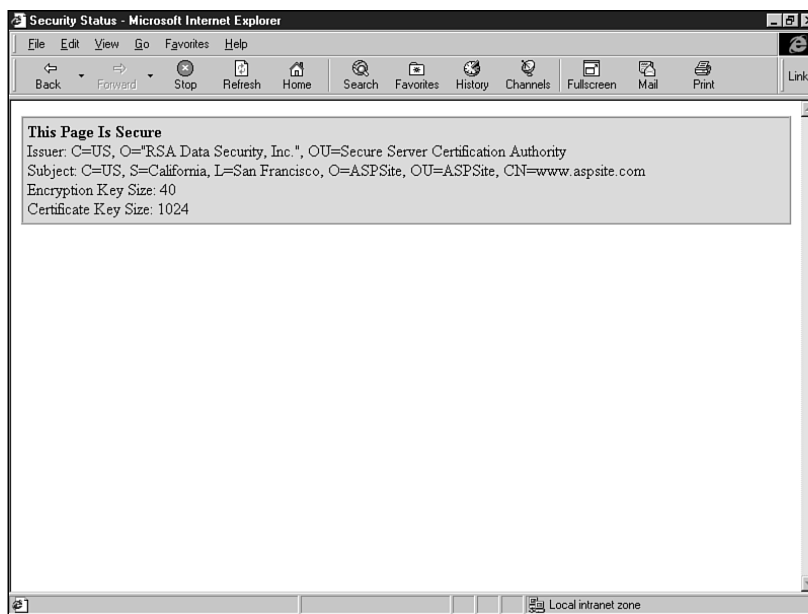
כאשר מבקשים נתוני כרטיס אשראי, מומלץ לספק גם גירסה מאובטחת וגם גירסה לא מאובטחת של טופס איסוף המידע. אף על פי שפרוטוקול SSL ישן כמעט כמו Netscape גירסה 1.0, באופן מוזר ישנם דפדפנים אשר עדיין אינם תומכים בו. כדי לאפשר למשתמשים לבחור בין שתי גרסאות לאותו דף, פשוט ספק שני קישורים שונים:

```
<A HREF="https://www.yourdomain.com/pay.asp">Pay Now (secure version)</A>
<A HREF="http://www.yourdomain.com/pay.asp">Pay Now (unsecure version)</A>
```

בתוך הדף pay.asp עצמו, באפשרותך לבדוק האם הבקשה לדף התבצעה באמצעות SSL או לא. אם המשתמש ביקש את הדף בצורה לא מאובטחת, אולי תרצה להזהירו. להלן דוגמה לשגרה המאבחנת ומדווחת על מצב האבטחה של דף ASP (ראה תרשים 23.4):

```
<%
SUB SecurityStatus
    IF Request.ServerVariables( "SERVER_PORT_SECURE" ) <> "1" THEN
        %>
        <TABLE width=100% cellpadding=4 cellpacing=0
        bgcolor="lightyellow" border=1 align=center>
        <TR>
            <TD><B>This Page Is Not Secure</B></TD>
        </TR>
        </TABLE>
    ELSE
        %>
        <TABLE width=100% cellpadding=4 cellpacing=0
        bgcolor="lightgreen" border=1 align=center>
        <TR>
            <TD>
```

```
            <B>This Page Is Secure</B>
            <BR>Issuer:
            <%=Request.ServerVariables( "CERT_SERVER_ISSUER" )%>
            <BR>Subject:
            <%=Request.ServerVariables( "CERT_SERVER_SUBJECT" )%>
            <BR>Encryption Key Size:
            <%=Request.ServerVariables( "CERT_KEYSIZE" )%>
            <BR>Certificate Key Size:
            <%=Request.ServerVariables( "CERT_SECRETKEYSIZE" )%>
            </TD>
        </TR>
        </TABLE>
        <%
    END IF
END SUB
%>
```




**תרשים 23.4** מצב האבטחה

שגרה זו משתמשת במשתנה השרת SERVER_PORT_SECURE מהאוסף ServerVariables כדי לקבוע אם הבקשה לדף בוצעה תוך שימוש ב-SSL. אם הבקשה לדף נעשתה באמצעות SSL, מדווחים מספר מאפיינים של הקשר המאובטח, כמו למשל האישור, אורך מפתח ההצפנה ושמו של מנפיק האישור.

### הערה:

כדי ללמוד עוד על האוסף ServerVariables, ראה פרק 2.

# הגנת האתר באמצעות סיסמה

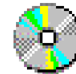
נניח שעליך ליצור אתר אינטרנט בו מחויבים המשתמשים להירשם לפני שיוכלו לקבל גישה לאזורים מוגני סיסמה. למשל, אולי תפתח חנות מקוונת בה חייבים המשתמשים להירשם לפני שיוכלו לקנות מוצר כלשהו. או שאולי תיצור אתר חיפוש עבודה בו חייבים החברים להיכנס למערכת (Login) לפני שיוכלו לראות את רשימת הצעות העבודה הקיימות.

תוכל לנקוט בשתי גישות כדי להגן על אתר באמצעות סיסמה בשימוש עם ASP. תוכל ליצור מערכת סיסמאות מותאמת, באמצעות שימוש במסד נתוני אבטחה, או שתוכל להשתמש באבטחת Windows NT.

## שימוש במסד נתוני אבטחה

פתרון אחד לבעיית אבטחת אתר אינטרנט באמצעות סיסמה היא שימוש במסד נתוני אבטחה. לפני שמשתמש יוכל לגשת לאזור מוגבל, שמו והסיסמה שלו נבדקים מול טבלת מסד נתונים. בשני הסעיפים הבאים תלמד כיצד ליצור טופס רישום וקובץ INCLUDE סטנדרטי, בו תוכל להשתמש כדי להגן על חלקים שונים באתר שלך.

### טופס הרישום



טופס הרישום משמש כדי לאסוף שלושה חלקי מידע: שם המשתמש, סיסמה והעדפות Cookie (ראה תרשים 23.5). ההחלטה אם להשתמש או לא להשתמש ב-Cookie לצרכי זיהוי חייבת להינתן למשתמש, כיון שישנם מצבים בהם עלול להיות מסוכן לזהות משתמש באופן אוטומטי באמצעות שימוש ב-Cookie. אם יותר מאדם אחד משתמש בדפדפן, ה-Cookie עלול לאמת את האדם הלא נכון (לדוגמה, אם האדם משתמש בדפדפן באינטרנט-קפה). טופס הרישום נכלל בתדפיס 23.1 (register.asp הנמצא בתקליטור).

Username:
Password:
Automatically remember me with a cookie.
Register

**תרשים 23.5** טופס הרישום

```
<%
function checked( byVal val1, byVal val2 )
    if val1 = val2 then checked = " checked"
end function

username = TRIM( Request( "username" ) )
password = TRIM( Request( "password" ) )
remember = Request( "remember" )
%>
<HTML>
<HEAD><TITLE>Register</TITLE></HEAD>
<BODY>
<FONT FACE="Arial">Register</FONT>
<P>
<CENTER>
<TABLE WIDTH=300 CELLPADDING=10 CELLSPACING=0
 BGCOLOR="#eeeeee" BORDER=1>
<TR>
    <TD>
    <FORM METHOD="post" ACTION="register2.asp">
    <INPUT NAME="formscript" TYPE="hidden"
        VALUE="register.asp">
    Username:
    <INPUT NAME="username" SIZE=20 MAXLENGTH=20
        VALUE="<%=Server.HTMLEncode( username )%>">
    <INPUT NAME="username_req" TYPE="hidden"
        VALUE="You must enter a username">
    <BR>Password:
    <INPUT NAME="password" SIZE=20 MAXLENGTH=20
        VALUE="<%=Server.HTMLEncode( password )%>">
    <INPUT NAME="password_req" TYPE="hidden"
        VALUE="You must enter a password.">
    <BR><INPUT NAME="remember" TYPE="checkbox"
        VALUE="1" <%=checked( remember, "1" )%>>
    <SMALL>Automatically remember me with a cookie.</SMALL>
    <P><INPUT TYPE="submit" VALUE="Register">
    </FORM>
    </TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

עבודת הרישום העיקרית מתרחשת כאשר משתמש שולח את טופס הרישום ומגיע לקובץ register2.asp (ראה תדפיס 23.2). הקובץ register2.asp, משתמש בקובץ ה-INCLUDE של validateForm אשר תואר בפרק 3, כדי לוודא שהמשתמש הקליד גם שם משתמש וגם סיסמה. בשלב הבא, מבוצעת שגרת SQL

מאוחסנת בשם addUser, כדי להוסיף את נתוני הרישום החדשים לטבלה בשם Users. אם הטבלה מכילה כבר את שם המשתמש, מוחזרת מהשגרה שגיאה, על ידי שימוש בפרמטר החזרה (דבר זה מונע משני משתמשים להשתמש באותו שם משתמש). לבסוף, התסריט מוסיף cookie עם שם המשתמש, ואם המשתמש בחר כך, Cookie נוסף עם סיסמת המשתמש.

הנה הצהרות SQL המשמשות ליצירת טבלת המשתמשים, המכילה את נתוני ההרשמה:

```
CREATE TABLE users (
    userID int IDENTITY (0, 1) NOT NULL ,
    username varchar (20) NOT NULL ,
    password varchar (20) NOT NULL ,
    remember bit NOT NULL ,
    secretkey int NOT NULL default RAND() * 1000
)
GO

grant all on users to public
```

**הערה:**

קובץ אצווה (Batch File) של SQL ליצירת טבלת המשתמשים וכל השגרות המאוחסנות המתוארות בסעיף זה, נמצא בתקליטור תחת השם security.sql.

שים לב שהטבלה כוללת עמודת זיהוי עבור ID ייחודי של משתמש. עמודת הזיהוי מוסיפה לכל משתמש מספר זיהוי ייחודי, באופן אוטומטי. הטבלה מכילה גם עמודה בשם secretkey, המייצרת מספר אקראי באופן אוטומטי. מטרת עמודה זו תוסבר בסעיף הבא.

הצהרות SQL הבאות משמשות ליצירת השגרה המאוחסנת addUser של SQL:

```
create procedure addUser
(
    @username varchar(20),
    @password varchar(20),
    @remember bit
)
AS
if exists( select username from users
        where username = @username )
    begin
        return( 10 )
    end
```

```
insert users
    (
    username,
    password,
remember
    ) VALUES (
    @username,
    @password,
    @remember
    )

GO
grant all on addUser to public
```

ההצהרה If exists משמשת לבדיקה אם שם משתמש מסוים כבר קיים. אם כן, יוצאת השגרה המאוחסנת עם קוד חוזר בערך של 10. אחרת, נתוני הרישום החדשים מוכנסים לטבלה Users.

**תדפיס 23.2** הוספת רישום משתמש חדש - REGISTER2.ASP

```
<!-- #INCLUDE virtual="/chapter3/validateform.asp" -->
<!-- #INCLUDE virtual="/adovbs.inc" -->
<%
username = TRIM( Request( "username" ) )
password = TRIM( Request( "password" ) )
remember = Request( "remember" )

' Ready Database Connection
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"

' Convert Remember To Bit
if remember = "" then
    remember = 0
else
    remember = 1
end if

' Add New User
Set CMD = Server.CreateObject( "ADODB.Command" )
CMD.ActiveConnection = Con
CMD.CommandType = adCmdStoredProc
CMD.CommandText = "addUser"
CMD.Parameters.Append CMD.CreateParameter
    ➥ ( "returnStatus",adInteger, adParamReturnValue )
CMD.Parameters.Append CMD.CreateParameter
    ➥ ( "username",adVarChar, adParamInput, 20, username )
CMD.Parameters.Append CMD.CreateParameter
    ➥ ( "password",adVarChar, adParamInput, 20, password )
```

```
CMD.Parameters.Append CMD.CreateParameter
    ➥ ( "remember",adBoolean, adParamInput,, remember )
CMD.Execute
returnStatus = cINT( CMD( "returnStatus" ) )
CMD.ActiveConnection = Nothing
SET CMD = Nothing

' Check For Name Already Registered
if returnStatus = 10 then
    errorMSG = "The username " & username & " is already registered."
    errorForm
end if

'  Add Cookies
    Response.Cookies( "username" ) = username
    Response.Cookies( "username" ).Expires = "July 31, 2000"
    Response.Cookies( "username" ).Domain =
      ➥ Request.ServerVariables( "SERVER_NAME" )
    Response.Cookies( "username" ).Path = "/"
    Response.Cookies( "username" ).Secure = FALSE
if remember = "1" then
    Response.Cookies( "password" ) = password
    Response.Cookies( "password" ).Expires = "July 31, 2000"
    Response.Cookies( "password" ).Domain =
      ➥ Request.ServerVariables( "SERVER_NAME" )
    Response.Cookies( "password" ).Path = "/"
    Response.Cookies( "password" ).Secure = FALSE
end if
%>
<HTML>
<HEAD><TITLE>Register</TITLE></HEAD>
<BODY>
<FONT FACE="Arial">Thank you for registering!</FONT>
</BODY>
</HTML>
```

## Security INCLUDE File

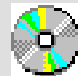
בכל דף עליו אתה מעוניין להגן באמצעות סיסמה, יהיה עליך לכלול את Security INCLUDE File בתחילת הדף, על ידי שימוש בהפניית #INCLUDE של ASP. קובץ זה ימנע ממשתמש לצפות בתוכן הדף, אלא אם המשתמש נכנס למערכת (ראה תדפיס 23.3).

```
<%
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'  Security INCLUDE file
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
sub requestPassword( byVal MSG )
    %>
    <HTML>
    <HEAD><TITLE>Login</TITLE></HEAD>
    <BODY>
    <FONT FACE="Arial">Login</FONT>
    <CENTER>
    <%=MSG%>
    <TABLE WIDTH=400>
    <TR>
        <TD>
        <FORM METHOD="post"
        ACTION="<%=Request.ServerVariables( "SCRIPT_NAME" )%>">
        username:
        <INPUT NAME="username" SIZE=20
        MAXLENGTH=20 VALUE="<%=Server.HTMLEncode( username )%>">
        <BR>password:
        <INPUT NAME="password" SIZE=20
        MAXLENGTH=20 VALUE="<%=Server.HTMLEncode( password )%>">
        <P><INPUT TYPE="submit" VALUE="Login">
        </FORM>
        </TD>
    </TR>
    <TR>
        <TD><A HREF="register.asp">Click here to register</A></TD>
    </TR>
    </TABLE>
    </CENTER>
    </BODY>
    </HTML>
    <%
    response.End
end sub

userKey = TRIM( Request( "k" ) )
username = TRIM( Request( "username" ) )
password = TRIM( Request( "password" ) )

' Ready Database Connection
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
```

```
if userKey <> "" then
    divider = INSTR( userkey, "p")
    userID = LEFT( userKey, divider - 1 )
    secretKey = RIGHT( userKey, LEN( userKey ) - divider )
    Set CMD = Server.CreateObject( "ADODB.Command" )
    CMD.ActiveConnection = Con
    CMD.CommandType = adCmdStoredProc
    CMD.CommandText = "checkKey"
    CMD.Parameters.Append CMD.CreateParameter
      ↳ ( "returnStatus",adInteger, adParamReturnValue )
    CMD.Parameters.Append CMD.CreateParameter
      ↳ ( "userID",adInteger, adParamInput, , userID )
    CMD.Parameters.Append CMD.CreateParameter
      ↳ ( "secretKey",adInteger, adParamInput, , secretKey )
    CMD.Parameters.Append CMD.CreateParameter
      ↳ ( "newsecretKey",adInteger, adParamOutput  )
    CMD.Execute
    returnStatus = cINT( CMD( "returnStatus" ) )
    newsecretKey = CMD( "newsecretKey" )
    CMD.ActiveConnection = Nothing
    SET CMD = Nothing
    if returnStatus <> 0 then
      ↳ requestPassword "You must login to access this page.<P>"
    k = "k=" & userID & "p" & newsecretKey
    formK = "<input name=""k"" type=""hidden"" value="""
      ↳ & userID & "p" & newsecretKey & """>"
else
    if username <> "" and password <> "" then
    Set CMD = Server.CreateObject( "ADODB.Command" )
    CMD.ActiveConnection = Con
    CMD.CommandType = adCmdStoredProc
    CMD.CommandText = "checkPassword"
    CMD.Parameters.Append CMD.CreateParameter
      ↳ ( "returnStatus",adInteger, adParamReturnValue )
    CMD.Parameters.Append CMD.CreateParameter
      ↳ ( "username",adVarchar, adParamInput, 20, username )
    CMD.Parameters.Append CMD.CreateParameter
      ↳ ( "password",adVarchar, adParamInput, 20, password )
    CMD.Parameters.Append CMD.CreateParameter
      ↳ ( "userID",adInteger, adParamOutput  )
    CMD.Parameters.Append CMD.CreateParameter
      ↳ ( "newsecretKey",adInteger, adParamOutput  )
    CMD.Execute
    returnStatus = cINT( CMD( "returnStatus" ) )
    userID = CMD( "userID" )
    secretKey = CMD( "newsecretKey" )
    CMD.ActiveConnection = Nothing
    SET CMD = Nothing
```

```
    if returnStatus = 20 then requestPassword
      ↳ "You did not enter a registered user name.<P>"
    if returnStatus = 10 then requestPassword
      ↳ "You did not enter a valid password.<P>"
    k = "k=" & userID & "p" & secretKey
    formK = "<input name=""k"" type=""hidden"" value="""
      ↳ & userID & "p" & secretKey & """>"
    else
    requestPassword "You must login to access this page.<P>"
    end if
end if
%>
```

### הערה:



התקליטור המצורף לספר זה כולל שני קבצים לדוגמה, securepage.asp ו- securepage2.asp, הממחישים כיצד יכול Security INCLUDE File לשמש להגנה על דפי ASP רבים באמצעות סיסמה.

אם משתמש מבקש דף המכיל Security INCLUDE File, יוצג טופס שיבקש את שם המשתמש והסיסמה הרשומים. אם ימולאו שם משתמש וסיסמה הם ייבדקו מול טבלת מסד הנתונים Users באמצעות השגרה המאוחסנת checkpassword (ראה תדפיס 23.4).

**תדפיס 23.4** בדיקת סיסמת המשתמש - checkpassword.asp

```
create procedure checkpassword
(
    @username varchar( 20 ),
    @password varchar( 20 ),
    @userID INT OUTPUT,
    @newsecretKey INT OUTPUT
)
AS
select @userID = userID from Users
    where username = @username
    and password = @password

if @userID is not NULL
    begin
        select @newsecretKey = RAND() * 1000
        update Users set secretKey = @newsecretKey
        where userID = userID
    end
    else
    begin
        if exists( select userID from Users
            where username = @username )
```

```
        begin
            return( 10 )
        end
        else
        begin
            return( 20 )
        end
    end

GO

grant all on checkpassword to public
```

השגרה המאוחסנת checkpassword מוודאת ששם המשתמש והסיסמה שסופקו אכן קיימים בטבלה Users. אם סופקו שם משתמש וסיסמה רשומים, השגרה המאוחסנת מייצרת secretKey חדש עבור המשתמש.

Security INCLUDE File בונה userKey בכל פעם שמשתמש מקבל גישה לדף מאובטח. userKey מכיל שילוב של userID ו-secretKey. לדוגמה, userKey עבור משתמש בעל userID 12 ו- secretKey 512 ייראה בערך כך:

12p512

מהי מטרתו של userKey? אם מישהו מקבל גישה למספר רב של דפי אינטרנט המוגנים בסיסמה, לא מומלץ להעביר את שם המשתמש והסיסמה בכל גישה לדף כלשהו. הבעיה היא שהסיסמה תוצג בסרגל הכתובת של חלון הדפדפן. כל מי שיציץ מעבר לכתפו של המשתמש יוכל לראות את הסיסמה הסודית.

במקום להעביר את שם המשתמש והסיסמה לכל הדפים מוגני הסיסמה, עליך להעביר את userKey. בכל פעם שהמשתמש ניגש לדף מוגן סיסמה נוצר userKey חדש, כך שגם אם מישהו מצליח לראות לרגע את userKey, הוא ישתנה ברגע שתתבצע בקשה לדף מוגן חדש.

לאחר שמיוצר userKey, תוכל לכלול אותו בכל הקישורים או הטפסים בגוף הדף, באמצעות שימוש במשתנה K או במשתנה formK. לדוגמה, כדי להעביר את userKey כחלק מקישור, השתמש בתחביר הבא:

```
<A HREF="securepage2.asp?<%=k%>">Next Page</A>
```

כדי להעביר את userKey כחלק מטופס, השתמש במשתנה formK. להלן דוגמה לטופס המכיל את userKey :

```
<FORM method="post"  action="securepage2.asp">
<%=formK%>
<input type="submit"  value="Next Page">
</FORM>
```

Security INCLUDE File משתמש בשגרה המאוחסנת checkKey כדי לבדוק האם ה-userKey בר תוקף (ראה תדפיס 23.5). השגרה המאוחסנת checkKey משתמשת בהצהרה If exists של SQL כדי לקבוע האם ה-userID וה-secretKey שהועברו לשגרה המאוחסנת אכן קיימים בטבלה. בנוסף, השגרה המאוחסנת יוצרת בצורה אוטומטית מפתח סודי אקראי.

**תדפיס 23.5** בדיקת מפתח אקראי - checkKey

```
create procedure checkKey
(
    @userID int,
    @secretKey int,
    @newsecretKey int OUTPUT
)
AS

if exists( select userID from Users
        where userid = @userID
        and secretKey = @secretKey )
    begin
        select @newsecretKey = RAND() * 1000
        update Users set secretKey = @newsecretKey
            where userID = @userID
    end
    else
    begin
        return( 10 )
    end
```

כאשר משתמש נרשם למערכת תוך שימוש בטופס הרישום שתואר בסעיף קודם, עומדת בפניו האפשרות לזיהוי אוטומטי באמצעות Cookie. כיון ש-Security INCLUDE File אינו מאזכר באופן מפורש את האוסף Cookies, בוודאי אתה תמה כיצד, אם כן, נתמכת תכונה זו.

התשובה היא שכאשר משתמשים ב-("Request("password במקום ב-("Request.Form("password או ב-("Request.QueryString("password, מבוצע חיפוש בכל האוסף Request. כיון שהאוסף Cookies הוא חלק מהאוסף Request, תמצא ("Request("password את הסיסמה, אפילו אם זו הועברה לדף באמצעות Cookie.

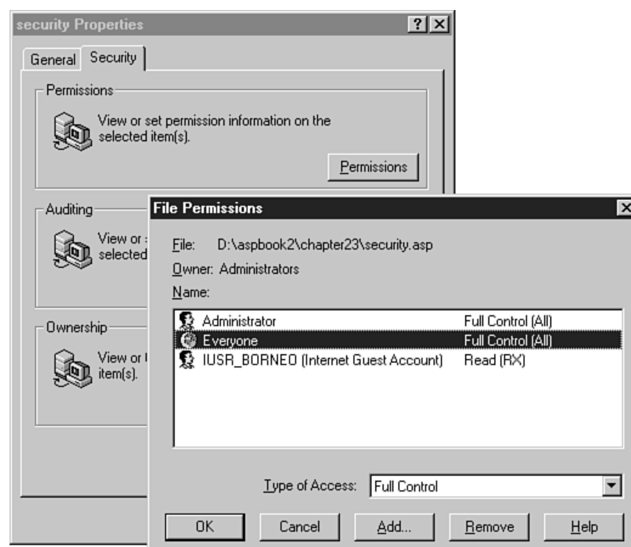
## שימוש באבטחת Windows NT

בחלק אחרון זה של הפרק, תלמד שיטה שנייה להגנת סיסמה על קבצים וספריות באתר שלך. חלק זה יסביר כיצד לשלב את אבטחת האתר שלך עם אבטחת מערכת ההפעלה Windows NT. תלמד כיצד להגן על דפי ASP באמצעות שימוש בחשבונות משתמש, חשבונות קבוצה והרשאות הקבצים של Windows NT.

**הערה:**

כדי להשתמש באבטחת Windows NT, אתה חייב להשתמש במערכת ההפעלה Windows NT עם מערכת הקבצים NTFS. אם שרת האינטרנט שלך המותקן בשרת המשתמש במערכת קבצים מסוג FAT, תוכל להמיר את הערכת הקבצים ל-NTFS באמצעות הפקודה CONVERT (ראה הערך convert בעזרה של Windows NT). לאחר המרה ל-NTFS לא תוכל להמיר בחזרה ל-FAT.

תחת מערכת הקבצים NTFS, יש לכל הקבצים והספריות ערכת הרשאות המקושרת אליהם. הרשאות אלו קובעות מי יכול לעשות דברים כגון קריאה, הפעלה או מחיקה של קובץ. כדי לראות הרשאה עבור קובץ, לחץ לחיצה ימנית על שם הקובץ, מתפריט הקיצור בחר Properties, בחר בכרטיסיה Security ולחץ על Permissions. תופיע תיבת הדו-שיח בתרשים 23.6.




**תרשים 23.6** תיבת הדו-שיח **File Permissions**.

תיבת הדו-שיח File Permissions מציגה רשימה של חשבונות משתמשים וקבוצות, ביחד עם הרשאות הגישה לקבצים המקושרות אליהן. לדוגמה, בתרשים 23.6, נראה חשבון המשתמש Administrator כחשבון שיש לו Full Control על הקובץ. משמעות הדבר היא, שאם תיכנס למערכת באמצעות שימוש בחשבון Administrator, יהיו לך ההרשאות לעשות כל שברצונך עם הקובץ, כולל למחוק אותו.

תוכל לשנות את הרשאות הגישה לכל חשבון משתמש או קבוצה על ידי לחיצה על הרשומה ובחירה בהרשאות החדשות מבין לחצני האפשרויות. כדי למנוע מכל מי שנכנס למערכת באמצעות החשבון Administrator למחוק לדוגמה קובץ מסוים, פשוט הסר את ההרשאה Delete של החשבון Administrator לקובץ זה.

כמו כן, תוכל להסיר לחלוטין חשבון משתמש או קבוצה, כך שלא תהיה להם הרשאת גישה כלשהי לקובץ. כדי לעשות זאת, בחר את חשבון המשתמש או הקבוצה ולחץ Remove. כדי להוסיף חשבון משתמש או קבוצה, לחץ Add.

**הערה:**

כדי ליצור חשבונות משתמש או קבוצה חדשים, תוכל להשתמש בתוכנית השירות Windows NT User Manager. בנוסף, תוכל לעבוד עם חשבונות משתמשים וקבוצות בתוך דפי ASP, תוך השימוש ב- Active Directory Service Interfaces (ADSI). ראה פרק 11, כדי ללמוד כיצד לעשות זאת.

בכל פעם שמתבצעת בקשה לדף מהאתר שלך, יקבל המבקש גישה לקובץ תחת חשבון משתמש כלשהו. כברירת מחדל, חשבון משתמש זה הוא IUSR_computername. לדוגמה, אם שם המחשב שלך הוא Plato, שמו של חשבון משתמש האינטרנט האנונימי יהיה IUSR_Plato.

בדיוק כמו בחשבונות אחרים, תוכל לקבוע את הרשאות הגישה לקובץ IUSR_computername. כדי למנוע ממשתמש אנונימי מלקבל גישה מהאינטרנט לקובץ המאוחסן בשרת שלך, תוכל למנוע מהמשתמש IUSER_computername את ההרשאה Read עבור הקובץ. כמו כן, ביכולתך למנוע ממשתמש אינטרנט לגשת לכל קובץ בספריה כלשהי, על ידי שינוי הרשאת חשבון המשתמש עבור הספריה.

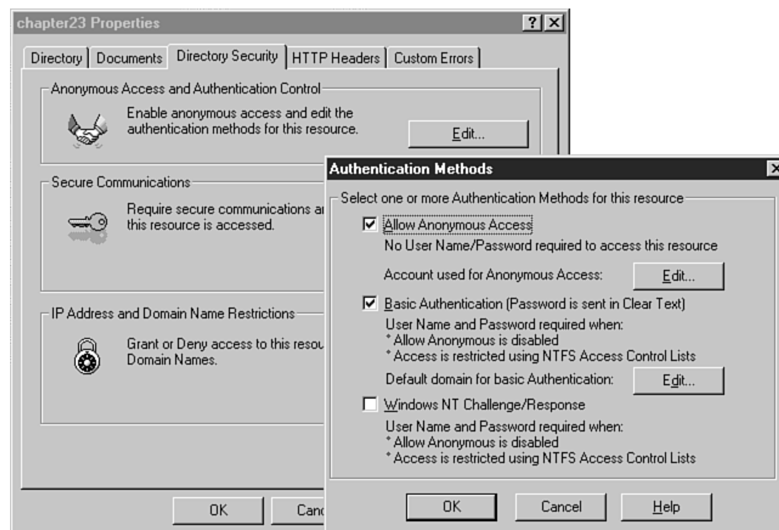
אם משתמש אנונימי מנסה לראות דף ASP ללא הרשאות מספיקות, תופיע תיבת דו-שיח המבקשת ממנו לבצע כניסה למערכת (Login). תוך שימוש בתיבת דו-שיח זו, יכול המשתמש האנונימי לזהות את עצמו כמשתמש מסוים ב-Domain שלך. לאחר שהמשתמש נכנס למערכת ואינו משתמש יותר בחשבון האנונימי IUSER_computername, הוא ימשיך להיות מזוהה על ידי החשבון האישי שלו לכל משך הביקור הנוכחי שלו באתר.

שימוש בחשבונות משתמש, קבוצות והרשאות קבצים של Windows NT, מאפשר לך ליצור רמת אבטחה דקת אבחנה לאתר שלך. תוכל ליצור כמה חשבונות משתמשים וקבוצות שאתה רוצה לפי צרכיך (Domain, יכול לתמוך באלפי חשבונות משתמשים וקבוצות), ותוכל לקבוע את ההרשאות לכל קובץ וספריה, על בסיס אינדיבידואלי.

## שיטות אימות

בסעיף הקודם, למדת כיצד לקבוע הרשאות עבור גישה לקבצים וספריות במערכת ההפעלה Windows NT. ניתן לשלוט בגישה גם באמצעות Internet Service Manager. תוכל לקבוע את הגישה עבור קבצים בודדים, ספריות, או אפילו אתרי אינטרנט שלמים.

כדי להשתמש ב-Internet Service Manager לשליטה בגישה לקובץ, ספריה או אתר אינטרנט, פתח את גיליון המאפיינים עבור האובייקט ובחר בכרטיסיה Directory Security או File Security. כעת, לחץ על הלחצן Edit תחת Anonymous Access ו-Authentication Control (ראה תרשים 23.7).




**תרשים 23.7** סטטוס האבטחה

כאשר משתמשים ניגשים לאתר שלך מבלי לבצע כניסה למערכת, הם מקבלים גישה אנונימית לאתר. כפי שתואר בסעיף הקודם, כברירת מחדל חשבון זה הוא IUSER_computername. כדי למנוע גישה ממשתמשי אינטרנט לא מאומתים, בטל את הסימון בתיבת הסימון Allow Anonymous Access. אם אתה מונע גישה אנונימית לספריה, תופיע תיבת דו-שיח לכניסה למערכת בכל פעם שמישהו ינסה לגשת לקובץ הנמצא בספריה זו.

אם תבחר לאפשר גישה אנונימית, תוכל לבחור את חשבון המשתמש אשר ישמש מבקרים לא מאומתים, על ידי הבחירה ב-Edit והקלדת שם המשתמש עבור חשבון המשתמש האנונימי. כברירת מחדל, חשבון המשתמש האנונימי הוא החשבון IUSR_computername. בכל אופן, באפשרותך לבחור בכל חשבון משתמש שתרצה.

אם האפשרות Enable Automatic Password Synchronization נבחרה, חשבון המשתמש האנונימי וחשבון המשתמש המתאים לו ב-Windows NT יקבלו באופן אוטומטי את אותה הסיסמה. אם לא תבחר באפשרות זו, תצטרך לסנכרן את שתי הסיסמאות ב-Internet Service Manager וב-Windows NT User Manager.

IIS (Internet Information Server) תומך בשני סוגי גישה מאומתת. ניתן לאפשר Basic Authentication או Windows NT Challenge/Response, או את שניהם גם יחד. בחירה זו קובעת כיצד מועברים שם המשתמש והסיסמה ברשת האינטרנט, כאשר משתמשים נכנסים למערכת באמצעות תיבת הדו-שיח לכניסה למערכת.

לאחר שמשתמש נכנס למערכת, שם המשתמש והסיסמה נכללים ככותרת HTML בכל פעם שהמשתמש מבקש דפים מהאתר. שם המשתמש והסיסמה ימשיכו להיכלל בכל בקשת HTML, עד שהמשתמש יסגור את הדפדפן, או שיתבקש להזדהות שנית.

אם תבחר באפשרות Basic Authentication, שם המשתמש והסיסמה יישלחו בצורה לא מקודדת. לדוגמה, לאחר שמשתמש אומת באמצעות Basic Authentication, בקשת HTTP תיראה בערך כך:

```
GET /somedir/somepage.asp
Host: www.aspsite.com
HTTP/1.0
Connection: Keep-Alive
Athorization: Basic QWRtaW5pa#ryYYRjcjpudXQ5MHRyXWU
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
```

כותרת האישור מכילה את מידע האימות. בדוגמה זו, מילת המפתח Basic מציינת שמתבצע שימוש ב-Basic Authentication ומחרוזת הקידוד שבאה לאחר מילת המפתח מכילה, למעשה, את שם המשתמש והסיסמה.

**הערה:**

כברירת מחדל, כאשר משתמשים נכנסים למערכת באמצעות השימוש ב- Basic Authentication, הם חייבים לכלול שם Domain יחד עם שם המשתמש שלהם. לדוגמה, עליהם להיכנס באמצעות שימוש במחרוזת במבנה domain/username, ולא רק username. בדרך כלל, זה לא מה שתרצה. תוכל להשתמש בלחצן Edit שליד תיבת הסימון Basic Authentication, כדי לספק Domain ברירת מחדל, כך שמשתמשים לא יצטרכו לספק Domain בעת כניסתם למערכת.

**קידוד** (Uuencoding) מציע רמה מינימלית ביותר של חשאיות. פורץ מחשבים (האקר) יכול לפענח את הכותרת הקודמת באמצעות שימוש במספר רב של uuencoders/decoders זמינים (חפש ב-Yahoo! את הערך "uuencode" כדי לקבל רשימה של תוכניות שרות אלו). במילים אחרות, שם המשתמש והסיסמה בבקשה המוקדמת יכולים להיות גם טקסט פשוט. Basic Authentication הוא לא סוג אימות בטוח ביותר להעברת שמות משתמשים וסיסמאות.

IIS גם תומך בסוג אימות שני הנקרא Windows NT Challenge/Response (נקרא גם "אימות NTLM"). Windows NT Challenge/Response מציעה רמת אבטחה גבוהה הרבה יותר.

אם משתמש נכנס למערכת באמצעות Windows NT Challenge/Response, שם המשתמש והסיסמה אינם מועברים למעשה אף פעם ברשת האינטרנט. תהליך האימות של Windows NT Challenge/Response כולל את השלבים הבאים:

1. משתמש מבקש דף הדורש אימות Windows NT Challenge/Response.
2. השרת שולח בחזרה לדפדפן ערך אקראי.
3. הדפדפן שולח בחזרה לשרת שלושה ערכים: שם המשתמש, שם doamin ואסימון (Token). האסימון מכיל את הסיסמה, המוצפנת באמצעות פונקציית גיבוב (Hash Function), תוך שימוש שימוש בערך האקראי שסופק על ידי השרת.
4. השרת שולח את שם המשתמש, שם domain וערך אקראי ל-Domain Controller.
5. ל-Domain Controller יש מידע לגבי כל שמות המשתמשים והסיסמאות עבור ה-domain. הוא משתמש בעותק שלו של הסיסמה וערך אקראי כדי להצפין אסימון, המייצג את שם המשתמש ואת שם ה-domain. האסימון נשלח חזרה לשרת.
6. השרת משווה את האסימון שהתקבל מה-domain Controller (שלב 5) והאסימון שקיבל מהדפדפן (שלב 3). אם שני האסימונים זהים - המשתמש מאומת.

שים לב שבמשך כל מהלך ההידברות המסובך הזה, השרת אינו מקבל אף לא פעם אחת את סיסמת המשתמש באופן שאינו מוצפן. הסיסמה הלא מוצפנת אינה מועברת בין הדפדפן לשרת, או בין השרת ל-Domain Controller. לכן, אין אף שלב בו ניתן לגנוב את הסיסמה מהרשת.

לרוע המזל, לאימות Windows NT Challenge/Response יש שלוש מגבלות חמורות. קודם כל, רק Internet Explorer (גירסה 2.0 ומעלה) תומך בקוד המקור שלו בשיטת אימות זו. אם משתמשי האתר שלך משתמשים בדפדפן Netscape, הם לא יוכלו להשתמש באימות Windows NT Challenge/Response ללא טעינה והתקנה של תוכנה נוספת (Microsoft Authentication Proxy for Netscape Navigator, אשר זמינה להורדה חופשית באתר של Microsoft). אם עליך להשתמש בסוג אימות שנתמך בצורה נרחבת - אין לך ברירה אלא להשתמש ב-Basic Authentication.

שנית, סוג האימות Windows NT Challenge/Response אינו עובד עם שרתי Proxy. שוב, אם אתה משתמש בשרת Proxy, אין לך ברירה אלא להשתמש ב-Basic Authentication.

**הערה ל- Windows 2000:**

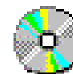
גירסה 5.0 של Internet Information Server תומך בסוג אימות נוסף, הנקרא Digest Authentication. כמו באימות Windows NT Challenge/Response, כאשר משתמשים ב- Digest Authentication, סיסמאות אינן נשלחות ברשת האינטרנט. אולם, שלא כמו באימות Windows NT Challenge/Response, סוג האימות Digest Authentication מאפשר עבודה דרך שרתי Proxy. Digest Authentication עובד עם Internet Explorer (גירסה 4.0 ומעלה).

לבסוף, סוג האימות Windows NT Challenge/Response אינו עובד כאשר יש צורך בגישה למשאבים ממספר שרתים. לדוגמה, אם יש לך דף ASP על שרת אחד החייב לקבל גישה למשאב בשרת אחר, כמו קובץ או קובץ MDB של Microsoft Access, אזי אימות Windows NT Challenge/Response ייכשל. מגבלה זו אינה קיימת אם IIS פועל ב-Domain Controller. כמו כן מגבלה זו אינה קיימת כאשר משתמשים ב-Basic Authentication.

ניתן לאפשר אימות Basic Authentication במקביל לאימות Windows NT Challenge/Response. אם הדפדפן אינו תומך ב-Windows NT Challenge/Response, הוא ישתמש אוטומטית ב-Basic Authentication.

## אחזור שמות משתמשים וסיסמאות

לאחר שמשתמש אומת, באפשרותך לאחזר את נתוני האימות של המשתמש. אם המשתמש אומת באמצעות שימוש ב-Basic Authentication, ביכולתך לאחזר גם את שם המשתמש וגם את הסיסמה. במקרה של אימות באמצעות Windows NT Challenge/Response, ניתן לאחזר רק את שם המשתמש.



התסריט הבא משתמש במשתנים מאוסף משתני השרת כדי לאחזר את שם המשתמש והסיסמה (נמצא בתקליטור תחת השם authinfo.asp) :

```
<%
authMethod = Request.ServerVariables( "AUTH_TYPE" )
username = Request.ServerVariables( "AUTH_USER" )
password = Request.ServerVariables( "AUTH_PASSWORD" )
%>
<HTML>
<HEAD><TITLE>Authentication Information</TITLE></HEAD>
</BODY>
<BR><B>Username:</B> <%=username%>
<BR><B>Password:</B> <%=password%>
<BR><B>Authentication method:</B> <%=authMethod%>
</BODY>
</HTML>
```

בדוגמה זו, מכיל משתנה השרת AUTH_TYPE את הערך BASIC או NTLM, תלוי בסוג האימות. המשתנה AUTH_TYPE מכיל את שם המשתמש. אם אתה משתמש באימות Windows NT Challenge/Response, משתנה זה יכיל גם את שם ה-domain. לבסוף, במקרה של Basic Authentication (אבל לא Windows NT Challenge/Response), מכיל המשתנה AUTH_PASSWORD את סיסמת המשתמש.

## אימות דינמי עם ASP

בכל פעם בה מבקש מישהו דף הדורש אימות, מגיב השרת על ידי שליחת קוד המצב 401 Not Authorized כתגובה. קוד מצב זה מודיע לדפדפן לפתוח את תיבת הדו-שיח לכניסה למערכת (Login Dialog-Box). תוכל להשתמש בשיטה זו כדי להכריח אימות בתוך תסריטי ASP.

נניח, לדוגמה, שברצונך להגן בסיסמה על דפים רק בשעות מסוימות במשך היום. דמיין לעצמך שהאתר שלך מארח מסמכים שברצונך להגביל את השימוש הפנימי בהם במשך שעות העבודה. אם מישהו ירצה לגשת למסמכים אלה בין השעה 8 בבוקר לשעה 6 בערב, הוא יהיה חייב לספק סיסמה. לעומת זאת, בשעות אחרות אתה מעוניין להפוך את המסמכים לזמינים לכל אחד באינטרנט.

באפשרותך להגן באמצעות סיסמה על דפי ASP בשעות מסוימות במשך היום, על ידי שימוש בתסריט הבא (נמצא בתקליטור, בקובץ בשם dynamicpassword.asp):

```
<%
if hour( time ) > 8 AND hour( time ) < 14  then
    username = TRIM( Request.ServerVariables( "AUTH_USER" ) )
    if username = "" then
        Response.Status = "401 Not Authorized"
        Response.addHeader "WWW-Authenticate", "Basic Realm=""YourDomain"""
        %>
        You cannot access this page until after 6:00pm.
        <%
    Response.End
    end if
end if
%>
<HTML>
<HEAD><TITLE>Business Document</TITLE></HEAD>
</BODY>
This is an interesting business document.
</BODY>
</HTML>
```

בתסריט זה, אם משתנה השרת AUTH_USER אינו מכיל ערך, והשעה היא במשך שעות העבודה, צריכה להופיע תיבת הדו-שיח בכניסה למערכת. שים לב שכותרת גם כן נכללת עם התגובה. הכותרת משמשת לציין את domain ברירת המחדל, כך שהמשתמש לא יצטרך למלא אותו בעצמו. עליך להחליף את "YourDomain" בשם ה-domain של Windows NT.

# סיכום

בפרק זה למדת כיצד לאבטח את אתר האינטרנט שלך. בחלק הראשון למדת כיצד להשתמש ב-SSL כדי להגן על פרטיות המידע שלך בעודו מועבר באינטרנט או ברשת המקומית. בהמשך הפרק למדת שתי שיטות להגנה על חלקים מהאתר באמצעות סיסמה. למדת כיצד ליצור Security INCLUDE File סטנדרטי, כדי להגן באמצעות סיסמה על האתר עם מסד נתוני אבטחה. כמו כן למדת כיצד לשלב את אבטחת האתר עם אבטחת Windows NT.

## פרק 24

# תחזוקת אתרי ASP

**בפרק זה:**

- **שימוש ב-Windows Scripting Host**
- **שימוש ב-Windows Scripting Host עם IIS Admin**
- **שימוש ב-Windows Task Scheduler**

תחזוקת אתר אינטרנט גדול עשויה להיות משימה משעממת ולגזול זמן רב. עליך לנטר בקביעות את האתר, כדי לוודא שכל חלקיו פועלים כשורה. בכל רגע נתון, ומכל סיבה שהיא, יכול שרת ה-IIS להפסיק לפעול, מסד הנתונים שלך עלול לחרוג מנפח הדיסק, או שמישהו יכול פשוט להחליט למשוך את כבל החשמל של השרת מהשקע בקיר.

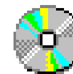
משתמשים מצפים מאתר אינטרנט ציבורי להיות זמין בכל רגע נתון במשך היום, בכל יום בשבוע. לאתרי אינטרנט אין חופשות או סוף שבוע פנוי. אם אין לך את המשאבים להושיב מישהו ליד שרת האינטרנט 24 שעות ביממה, עליך לפתח שיטות לניטור אוטומטי של האתר. בפרק זה תלמד כיצד לעשות זאת.

בחלקו הראשון של הפרק תלמד כיצד להשתמש ב-Windows Scripting Host (WSH) ליצירת תסריטים הפועלים בצורה בלתי תלויה בשרת האינטרנט. Windows Scripting Host יכול לשמש למיכון משימות ניהול וניטור שכיחות רבות. לדוגמה, ניתן להשתמש ב-Windows Scripting Host כדי לגבות באופן אוטומטי את קבצי השרת שלך על בסיס יומי, או לבדוק באופן אוטומטי את מצב האתר, על פי תזמון קבוע במשך היום.

חלקו השני של הפרק דן ב**מתזמן המשימות של Windows** (Windows Task Scheduler). מתזמן המשימות מספק **ממשק משתמש גרפי - GUI**, עבור הפקודה AT של DOS. תוכל להשתמש במתזמן המשימות כדי לתזמן הפעלת תסריטים של Windows Scripting Host.

# שימוש ב-Windows Scripting Host

Windows Scripting Host (WSH) הוא חלק משרת IIS, ומצורף בחבילת התוכנה Windows NT Option Pack. WSH הוא חלק אינטגרלי ממערכת ההפעלה Windows 2000. תוכל להשתמש ב-WSH כדי להפעיל תסריטי VBScript או Jscript, מתוך Windows או משורת הפקודה של DOS.



תסריט WSH נראה דומה מאוד לתסריט ASP. להלן תסריט פשוט, אשר מדפיס 52 פעמים את המילים "Hello World" (נמצא בתקליטור, בשם hello.vbs):

```
for i = 1 to 52
   Wscript.Echo "Hello World!"
next
```

אם תכתוב תסריט זה בפנקס הרשימות (Notepad) ותשמור אותו בשם hello.vbs, תוכל להפעיל אותו משורת הפקודה של DOS, על ידי כתיבת הפקודה cscript hello.vbs. סיומת הקובץ vbs. היא חשובה, כיון שהיא אומרת ל-WSH שאתה מנסה להפעיל תסריט VBScript. כדי להפעיל תסריט JavaScript אתה חייב להשתמש בסיומת js .

שים לב לשלושה דברים בנוגע לתסריט זה. ראשית, נעשה שימוש בהצהרות VBScript סטנדרטיות. WSH משתמש באותו מנוע תסריט כמו ASP. משמעות הדבר היא שניתן להשתמש ב-VBScript בתסריט WSH, בדיוק באותו אופן כמו ב-ASP (הדבר נכון גם לגבי JScript).

שנית, לא נעשה שימוש בתוחמי התסריט %> ו- <%. תוחמי תסריט אינם נחוצים, כיון שתסריט WSH מורכב אך ורק מתסריט. אינך יכול לשבץ קוד HTML בתסריט WSH, כפי שעשית בתסריטי ASP.

הדבר השלישי אליו יש לשים לב בתסריט, הוא השיטה להדפסת Hello World. בתסריט הקודם, משמשת השיטה ()Echo של האובייקט Wscript להדפסת ההודעה. אינך יכול להשתמש באובייקט Response כדי להדפיס את ההודעה כפי שהיית עושה בקובץ ASP. בעצם, אינך יכול להשתמש כלל באובייקטים הפנימיים של ASP מתוך תסריט WSH, כגון באובייקטים Application, Session או Request. מצד שני, אתה יכול להשתמש ברכיבי ActiveX כגון ADO, האובייקט File Access או CDO עבור NTS (בהמשך הפרק תראה כיצד לעשות זאת).

# הפעלת תסריטי WSH

ניתן להפעיל תסריט WSH (Windows Scripting Host) באחת משתי דרכים. אם אתה מעוניין שהתסריט ידפיס את התוצאות שלו בחלון שורת הפקודה של DOS, עליך להשתמש בפקודה cscript. לחילופין, תוכל לבחור לבצע את תסריט ה-WSH בסביבת Windows, על ידי שימוש בפקודה wscript. כאשר אתה מפעיל תסריט באמצעות הפקודה wscript, תוצאות התסריט מוצגות בתיבת דו-שיח של Windows. תרשים 24.1 מציג את התוצאות שהתקבלו מהפעלת התסריט hello.vbs עם הפקודה wscript. כדי לגרום לתיבת הדו-שיח להיעלם, עליך ללחוץ על לחצן OK 52 פעמים, לחיצה עבור כל הצגה של ההודעה !Hello World.

**תרשים 24.1** תיבת הדו-שיח מהפקודה Wscript

שתי הפקודות, cscript ו-wscript, מקבלות מספר פרמטרים אופציונליים שימושיים נוספים. תוכל לתמוך בכל סוג פלט מהתסריט, על ידי שימוש בפרמטר B//. כאשר אתה מפעיל תסריטי WSH מ**מתזמן המשימות** (Task Scheduler), מתואר בסעיף **שימוש**

**במתזמן המשימות של Windows**), עליך להשתמש בפרמטר זה כדי למנוע את הצפת מסך השרת שלך בתיבות דו-שיח פתוחות. כדי להפעיל תסריט wscript כמו התסריט hello.vbs עם הפרמטר B// כתוב:

wscript hello.vbs //B

אם התסריט hello.vbs יציג פלט כלשהו, הפלט ינופה.

פרמטר שימושי אופציונלי נוסף הוא הפרמטר T//. פרמטר זה מגדיר את משך הזמן בו יפעל התסריט למספר שניות קבוע מראש. פרמטר זה יעיל במיוחד כדי לשלוט בתסריטים היוצאים מכלל שליטה. לדוגמה, אם תסריט יילכד בלולאת WEND. . .WHILE, יוכל הפרמטר T// לאלץ את התסריט להסתיים לאחר מספר שניות מסוים. הפקודה הבאה מגדירה את משך פעולת התסריט hello.vbs לשתי שניות:

cscript hello.vbs //T:2

### הערה:

ניתן לבטל בכל זמן שהוא תסריט WSH פעיל, אשר הופעל באמצעות הפקודה cscript. הדבר נעשה באמצעות שימוש בשילוב המקשים Ctrl+C. צירוף מקשים זה סוגר את חלון ה-DOS. הפעלת צירוף המקשים Ctrl+C מקביל לצירוף Alt+F4 ביישום חלונאי.

טבלה 24.1 מונה את הפרמטרים הניתנים לשימוש כאשר מפעילים תסריטי WSH:

**טבלה 24.1** פרמטרים של WSH

| פרמטר | פעולה |
|---|---|
| //I | Interactive Mode - מצב אינטראקטיבי: דורש התערבות משתמש ואינו מונע שגיאות (ברירת המחדל). |
| //B | Batch Mode - מצב אצווה: אינו דורש התערבות משתמש ומונע שגיאות. |
| //T:nn | מספר השניות שבו רשאי התסריט לפעול. |
| //logo | הצגת לוגו בזמן פעולת התסריט (ברירת המחדל). |
| //nologo | מונע תצוגת לוגו בזמן פעולת התסריט. |
| //H:Cscript | מבצע רישום של Cscript כשפת ברירת המחדל להפעלת תסריטים. |
| //H:Wscript | מבצע רישום של Wscript כשפת ברירת המחדל להפעלת תסריטים. |
| //S | שומר את אפשרויות שורת הפקודה הנוכחיות. |
| //? | מציג עזרה לפרמטרים. |

בתוך תסריט WSH, ניתן לעצור את ביצוע התסריט על ידי שימוש בשיטה ()QUIT של האובייקט Wscript. לשימוש בשיטה ()Wscript.QUIT תוצאה זהה לזו המתקבלת

משימוש בשיטה Response.End מתוך דפי ASP. בדוגמה הבאה, לעולם לא תוצג ההודעה I am never seen! :

```
WScript.Echo "Hello! "
WScript.QUIT
WScript.Echo "I am never seen! "
```

# שימוש ברכיבי ActiveX מתוך WSH

כמעט בכל רכיב ActiveX בו ניתן להשתמש בדפי ASP, ניתן להשתמש גם בתסריט WSH. למשל, תוכל לבצע שאילתה למסד נתונים תוך שימוש ב-ADO, לשלוח דואר אלקטרוני תוך שימוש ב-CDO עבור CDO for NTS) NTS), או לגשת למערכת הקבצים תוך שימוש באובייקט הגישה לקבצים - File Access. כדי ליצור מופע של אובייקט ActiveX בתסריט WSH, עליך להשתמש ב-WScript.CreateObject. לדוגמה, התסריט הבא משתמש ב-ADO, כדי לאחזר ולהציג את שמות כל המחברים מטבלת המחברים בדוגמת מסד הנתונים Pubs הנכלל עם שרת SQL :

```
Set Con = WScript.CreateObject("ADOBE.Connection" )
Con.Open "DSN=MyDSN;UID=username;PWD=secret"
MySQL = "select au_fname, au_lname from authors"
Set RS = Con.Execute( MySQL )
while not RS.EOF
    wscript.echo RS("au_fname" ) & " " & RS("au_lname" )
    Rs.MoveNext
wend
```

שים לב שהתסריט נראה זהה לחלוטין לתסריט ASP רגיל, חוץ מאשר העובדה שכדי ליצור מופע של אובייקט ActiveX נעשה שימוש בשיטה WScript.CreateObject, במקום בשיטה Server.CreateObject, וכן נעשה שימוש בשיטה WScript.Echo, במקום ב-Response.Write, כדי להציג את הפלט.

**הערה:**

לא ניתן לכלול קבצים נוספים בתסריט WSH. אין הפניה השקולה להפניה #INCLUDE של ASP. זוהי מגבלה חמורה של WSH, ותקוותי הגדולה היא ש- Microsoft יחליטו לכלול שיטה שתעשה זאת בגירסה הבאה של WSH. אחת ההשלכות של מגבלה זו היא שאין כל שיטה לכלול את קובץ הקבועים של ADO (adovbs.inc) בתסריט WSH. כדי להשתמש בקבועים של ADO, עליך להעתיק קובץ זה לקובץ ה- WSH, או להשתמש בערכים לא מעובדים של הקבועים בתסריט שלך. אף אחת משיטות אלו אינה אלגנטית במיוחד.

# קבלת פרמטרים של שורת הפקודה

ניתן להפעיל תסריט WSH עם פרמטרים של שורת הפקודה. פרמטרים של שורת הפקודה מספקים לך שיטה להעברת ערכים לתסריט.לדוגמה, התסריט הבא שולח הודעת דואר אלקטרוני לכתובות שונות, בהסתמך על ערך פרמטר שורת הפקודה שהועבר לתסריט כאשר זה הופעל (את התסריט תמצא בתקליטור המצורף, בשם commandline.vbs):

```
' Sendmail.vbs
Set Args = WScript.Arguments
if Args.Count > 0 then
    email = Args( 0 )
    SET myMail = WScript.CreateObject("CDONTS.Newmail")
    myMail.From = "admin@yoursite.com"
    myMail.To = email
    myMail.Subject = "Hello from WSH!"
    myMail.Body = "This message was sent from WSH."
    myMail.Send
    SET myMail = Nothing
else
    WScript.Echo "You must enter an email address"
end if
```

תסריט זה משתמש במאפיין Argument של האובייקט WScript כדי לאחזר את הפרמטרים שהועברו לתסריט כאשר הופעל. תוכל להשתמש במאפיין Count, כדי לבדוק האם פרמטר כלשהו אכן הועבר בפועל. בדוגמה זו, כתובת הדואר האלקטרוני נקבעה לפרמטר הראשון אשר נשלח לתסריט. ההצהרה הבאה שולחת דואר אלקטרוני לכתובת billg@microsoft.com:

```
cscript sendmail.vbs billg@microsoft.com
```

# שימוש ב-WSH עם IIS Admin

אחד מהיישומים המועילים ביותר של WSH משמש להגדרה ולניטור אוטומטי של אתר IIS. ניתן להשתמש ב-WSH עם אובייקטים של IIS Admin כדי לטפל בכל אחד מהמאפיינים השונים של שרת האינטרנט. לדוגמה, תדפיס 24.1 מציג רשימה של כל אתרי האינטרנט שלך, ומצבם הנוכחי (תמצא את תוכן התדפיס בקובץ בשם showStatus.vbs בתקליטור המצורף):

**תדפיס 24.1** הצגת רשימת אתרי אינטרנט, ומצבם

```
Set ADsObj = GetObject( "IIS://localhost/W3SVC")
for each key in ADsObj
    if isNumeric( key.Name ) then
        wscript.echo "=============="
        wscript.echo "Server: " & key.ServerComment
        wscript.echo "State: " & showState( key.ServerState )
    end if
next
```

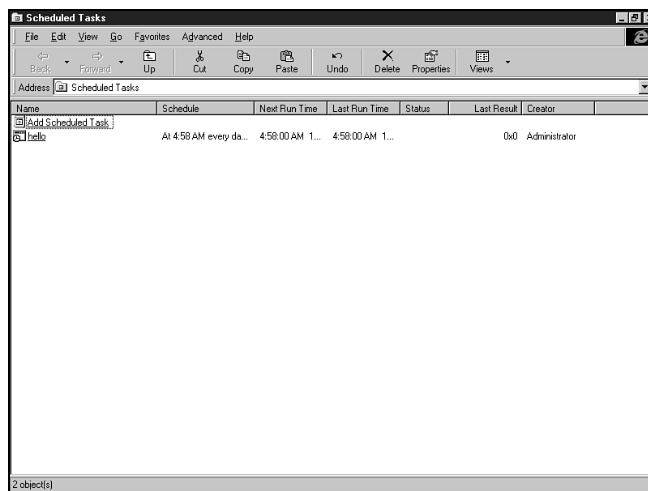
```
function showState( theState )
    select case theState
        case 1
            showState = "Starting..."
        case 2
            showState = "Started"
        case 3
            showState = "Stopping..."
        case 4
            showState = "Stopped"
        case 5
            showState = "Pausing..."
        case 6
            showState = "Paused"
        case 7
            showState = "Continuing"
    end select
end function
```

# שימוש במתזמן המשימות של Windows

מערכת ההפעלה Windows NT כוללת פקודה להפעלת משימות בזמנים מתוזמנים: הפקודה AT. לדוגמה, תוכל להשתמש בפקודה AT כדי להפעיל תסריט או תוכנית פעם ביום, או רק ביום ראשון. הפקודה AT עובדת בצורה יוצאת מהכלל, אך אין זה נחמד להשתמש בה בצורה הגסה שלה. אפשרות טובה יותר היא להשתמש **במתזמן המשימות של Windows** (Windows Task Scheduler), אשר מספק ממשק משתמש ידידותי יותר לפקודה AT (ראה תרשים 24.2).

מתזמן המשימות של Windows אינו נכלל ב-Windows NT Option Pack. מתזמן המשימות הוא רכיב נוסף של Internet Explorer. אם מתזמן המשימות אינו מותקן במחשב שלך (בדוק בתיקיה Administrative Tools שבתפריט Start), תוכל לטעון אותו על ידי ביקור בחלק Internet Explorer באתר של Microsoft בכתובת: http://www.microsoft.com/windows/ie.

השימוש במתזמן המשימות הוא ברור. כדי ליצור משימה חדשה, לחץ על הסמל Add Scheduled Task. לחיצה על סמל זה תפעיל אשף אשר ינחה אותך, צעד אחרי צעד, בתהליך יצירת משימה חדשה. תתבקש לבחור ביישום להפעלה ובמועדים בהם אתה מעוניין שהמשימה תופעל.



**תרשים 24.2** מתזמן המשימות של Windows.

אם אתה מעוניין לתזמן משימה כך שתופעל יותר מפעם ביום, נניח כל חמש דקות, יהיה עליך לפנות לכרטיסיה Properties של Advanced Schedule Options עבור משימה זו. כדי לפתוח את כרטיסיית המאפיינים בחר Open Advance Properties of This Task מאשף מתזמן המשימות, או לחץ לחיצה ימנית על משימה מתוזמנת, בחר Schedule, ולחץ על Advanced.

לאחר שתזמנת משימה, היא תופעל במועדים שציינת, אפילו כשמתזמן המשימות של Windows אינו פתוח על שולחן העבודה. מתזמן המשימות שומר יומן של פעילויות שהופעלו. יומן זה נשמר בקובץ טקסט הנקרא SchedLog.txt, הממוקם בתיקיה Winnt. ניתן לקרוא את היומן באמצעות פנקס הרשימות (Notepad) או על ידי בחירה ב-Advanced, View Log מתוך מתזמן המשימות.

מתזמן המשימות שימושי מאוד להפעלת תסריטי WSH. לדוגמה, אם אתה רוצה לבצע גיבוי של כל הקבצים בשרת האינטרנט שלך כל ערב, תוכל לתזמן את תסריט ה-WSH הבא להפעלה בכל ערב, בשעה 11:00:

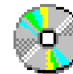
```
Set fs = WScript.CreateObject("Scripting.FileSystemObject" )
fs.CopyFolder "c:\inetpub\wwwroot", "c:\wwwrootbackup"
```

התסריט לדוגמה זה, בן שתי השורות, יוצר מופע של האובייקט File Access ומעתיק את כל הקבצים ותיקיות המשנה, מהתיקיה wwwroot לתיקיה בשם wwwrootbackup. תוכל להשתמש בתסריט דומה כדי לגבות או להעביר את קבצי היומן משרת האינטרנט שלך למחשב אחר על בסיס לילי (אם בשרת האינטרנט יש תעבורה רבה, קבצי היומן יכולים מהר מאוד למלא את כונן הדיסק הקשיח שלך).

יישום שימושי אחר של מתזמן המשימות הוא בדיקת מצב האתר. בפרק 16 למדת כיצד להשתמש ברכיב WinSock כדי לבקש דפי HTML מכל אתר אינטרנט. במקום להשתמש ברכיב WinSock כדי לאחזר דף מאתר אינטרנט אחר, ניתן להשתמש ברכיב כדי לנסות לאחזר דפים מאתר האינטרנט שלך בעצמו. אם אינך יכול לאחזר דף כלשהו, תדע שמשהו אינו כשורה.

כדי לבדוק מצב של דף, עליך לשבץ בתוכו מחרוזת נסתרת. לדוגמה, ניתן להחביא את המחרוזת "Good Page" בתוך הדף, על ידי שימוש בתווי הערה של HTML באופן הבא:

```
<! - -  Good Page  - ->
```



אם אתה מנסה לאחזר דף המכיל בתוכו מחרוזת נסתרת זו והמחרוזת לא יכולה להימצא, תוכל לדעת שיש שגיאה בדף, או ששרת האינטרנט אינו מתפקד כשורה. תדפיס 24.2 משתמש ברכיב w3 Sockets של WinSock, כדי לבקש את הדף default.asp ולבדוק את הימצאותה של המחרוזת הנסתרת (תמצא את התדפיס בקובץ בשם checksite.vbs בתקליטור המצורף).

**תדפיס 24.2** בדיקת אתר האינטרנט

```
YourSite = "www.yourdomain.com"
YourPage = "/default.asp"
CheckFor = "Good Page"

SET mySock = WScript.CreateObject("Socket.TCP")
mySock.Host =  YourSite & ":80"
theRequest = "GET " & YourPage & " HTTP/1.0" & vbCRLF
theRequest = theRequest & "Accept: image/gif, image/x-xbitmap, image/jpeg,
  ↳ image/pjpeg, */*" & vbCRLF
theRequest = theRequest & vbCRLF
mySock.Open()
mySock.SendLine theRequest
mySock.WaitForDisconnect()

if INSTR( mySock.Buffer, CheckFor ) > 0 then
    wscript.echo "Everything's fine!"
else
    wscript.echo "Could not retrieve " & YourPage
end if
mySock.Close()
```

תסריט זה מנסה לאחזר את הדף default.asp/ מהאתר www.yourdomain.com. אם הדף אוחזר, נסרק תוכן הדף לאיתור המחרוזת Good Page. אם לא ניתן לאתר את המחרוזת, מוצגת הודעת שגיאה.

במקום להציג בפשטות הודעת שגיאה, תוכל להשתמש ב-CDO עבור NTS ‏(CDO for NTS) כדי לשלוח הודעת דואר אלקטרוני באופן אוטומטי. אם תתזמן משימה זו באמצעות מתזמן המשימות כך שתתבצע כל חמש דקות, תוכל לדעת מייד אם אתר האינטרנט מפתח שגיאות או כישלונות, אפילו אם אתה בחופשה באיים הקריביים.

**טיפ:**

במקום לקבוע שהודעות שגיאה יישלחו לחשבון הדואר האלקטרוני שלך, תוכל להגדיר שהן תשלחנה לזימונית (ביפר) שלך. ספקי שירותי זימון מסוימים מאפשרים משלוח דואר אלקטרוני ישירות לזימונית, מה שמאפשר לך לשלוח הודעות טקסט ישירות לזימונית, באמצעות דואר אלקטרוני רגיל.

הרבה מכם, קוראי ספר זה, ימצאו ענין רב ב-WSH על מיגוון יכולותיו. למי שמעוניין יוכל למצוא מידע באתר Microsoft בכתובת www.microsoft.com/scripting.

לא כל היכולות של WSH נסקרו בפרק זה, למשל היכולת של הפונקציה SEndKeys אשר שולחת אירועים ליישום כלשהו פעיל במערכת ההפעלה, או פונקציה אחרת שנקראת Run אשר מפעילה יישום המשויך לשם קובץ.

# סיכום

בפרק זה למדת מספר שיטות נוספות בעלות ערך כדי לתחזק את אתר האינטרנט שלך. בחלק הראשון, למדת כיצד להשתמש ב-Windows Scripting Host ‏(WSH) כדי ליצור תסריטים שיכולים לשמש לביצוע פעולות הגדרה ותחזוקה. בחלקו האחר של הפרק למדת כיצד להשתמש במתזמן המשימות, כדי להפעיל תסריטים בזמנים מוגדרים באופן אוטומטי.

## פרק 25

# שיפור ביצועי דפי ASP

**בפרק זה:**

- **אופטימיזציה של דפי HTML**
- **אופטימיזציה של ביצועי מסד נתונים**
- **אופטימיזציה של דפי ASP**
- **תכנון קיבולת**

אתרי אינטרנט ציבוריים מצליחים נוטים לגדול מהר. כאשר האתר שלך עולה לראשונה לרשת האינטרנט, קבלת מאות מבקרים ביום יכולה להיות אירוע מרגש. בתוך כמה חודשים, בכל אופן, התעבורה באתר יכולה לצמוח לכמה אלפי או אפילו עשרות אלפי מבקרים ביום. אם לא תהיה מוכן לכך, התרגשותך הראשונית תהפוך במהרה לדאגה.

אם אתר האינטרנט שלך לא תוכנן לטיפול בתעבורה, דברים נוראים יקרו. תסריטי ה-ASP שלך יישחקו מרוב עבודה עד לקריסה מוחלטת. תתחיל לקבל הודעות על שגיאות פסק-זמן (Timeout) וקיפאון (Deadlock) משרת מסד הנתונים שלך. שרת האינטרנט יסרב להיענות לחיבורים (connections) של משתמשים נוספים, או יחזיר את הודעת השגיאה SERVER TOO BUSY. לבסוף, מוצף, סחוט מזיכרון וללא יכולות עיבוד נוספות, שרת האינטרנט שלך פשוט יקרוס.

עיצוב אתר אינטרנט שגודלו ניתן לשינוי היא משימה לא קלה. הקושי נובע ממספר גורמים הקובעים את ביצועי האתר. עליך לבחור בזהירות את החומרה המתאימה, להקים את חיבור הרשת הנכון, והכי חשוב, לעצב את תסריטי ה-ASP שלך בהתאם.

פרק זה מתמקד בבעיה האחרונה של אפשרויות שינוי גודל האתר. בפרק זה תלמד כיצד לשפר את ביצועי אתר ה-ASP שלך על ידי אופטימיזציה של דפי HTML, אופטימיזציה של הגישה למסד הנתונים, ואופטימיזציה של תסריטי ה-ASP שלך. כמו כן תלמד כיצד לבחון את אפשרויות שינוי גודל האתר שלך באמצעות שימוש ב-Microsoft InetMonitor.

# אופטימיזציה של דפי HTML

המוצר הסופי של תסריטי ASP הוא כמעט תמיד דפי HTML. דף HTML והתמונות שהוא מכיל הם בעצם הדברים אשר מועברים בפועל ברשת האינטרנט ומוצגים למשתמש. דף HTML שתוכנן בצורה לקויה יכול להשפיע בצורה רצינית על ביצועי אתר האינטרנט. בסעיפים הבאים תלמד כמה טיפים ליצירת דפי HTML יעילים יותר.

## אופטימיזציה של גרפיקה

הגורם הגדול היחידי הקובע את ביצועיו של דף HTML הוא המספר וגודל הגרפיקה שהוא מכיל. גרפיקה מאיטה את תצוגת הדף בשלוש דרכים. ראשית, גרפיקה אופיינית מכילה מידע רב. גודלו של קוד דף ה-HTML כולו יכול להתגמד לעומת גודלה של גרפיקה בודדת בדף. ככל שהגרפיקה מכילה יותר בתים (bytes) כך גם ייקח זמן רב יותר לגרפיקה לעבור ברשת ולהיות מוצגת בדפדפן.

שנית, הדפדפן חייב ליצור בקשה נפרדת עבור כל גרפיקה הנכללת בדף HTML. לדוגמה, אם דף HTML מכיל חמש תמונות, הדפדפן חייב לבצע חמש בקשות נפרדות מהשרת. מילוי בקשות אלו גוזל זמן עבודה רבה יותר בשרת.

לבסוף, אם הדפדפן אינו יודע את מימדי הגרפיקה, בניית הדף חייבת להיות מושהית או לחזור על עצמה שוב כאשר מתקבלת הגרפיקה בשלמותה. דפדפן אינו יכול לקבוע את תבנית הדף עד לאחר קבלת מידע על מימדי כל האלמנטים הנכללים בו.

אתרים בהם קיימת תעבורה כבדה, נמנעים בדרך כלל משימוש בגרפיקה. כאשר משתמשים בכל זאת בגרפיקה, היא עוברת אופטימיזציה זהירה ביותר. לדוגמה, דף הבית של Yahoo!, האתר בעל התעבורה הכבדה ביותר האינטרנט, מכיל כרגע רק תמונה אחת אשר אינה פרסומת. למרבה ההפתעה, גודלה של תמונה בודדת זו - כרזת הלוגו של Yahoo! - הוא פחות מ-5000 בתים (וזו תמונה גדולה).

עליך להימנע משימוש בגרפיקה באתר שלך, כל זמן שהדבר אפשרי. מפתחי אתרים מרגישים לעיתים לחץ מצד שותפים ומשקיעים לכלול גרפיקה באתר. אולם, ראה את אתרי האינטרנט המצליחים ביותר, כגון Yahoo!, eBay ו-Amazon, ומנה את מספר התמונות שהם מכילים. אפילו Wired, שנחשב פעם לאתר המכיל דפים המלאים בגרפיקה, הפך להיות אתר דל במספר התמונות שבו.

כאשר אתה מחליט לכלול גרפיקה, עליך לנהוג על פי הטיפים הבאים :

- בחר את פורמט הגרפיקה המתאים. שני פורמטים של גרפיקה הנתמכים כרגע ברשת האינטרנט הם GIF ו-JPEG. שני פורמטים אלה משתמשים באלגוריתמים שונים לדחיסה, כדי להפחית את גודל הגרפיקה. בדרך כלל, עליך להשתמש בפורמט JPEG עבור צילומים או תמונות אחרות המכילות גוונים רבים של צבעים. פורמט GIF מתאים יותר לשימוש בגרפיקה המכילה קווים חדים כמו סמלים (Icons) או כרזות פרסום (Banners). שימוש בפורמט גרפיקה שאינו מתאים יכול להשפיע בצורה דרמטית על גודל הגרפיקה.
- צמצם את גודל לוח הצבעים של התמונה. תמונה בפורמט GIF יכולה להשתמש בלוח צבעים המכיל 2 עד 256 צבעים. חשוב להבין כי תמונה יכולה להשתמש בלוח צבעים בעל 256 צבעים, אפילו אם התמונה עצמה משתמשת למעשה בשני צבעים בלבד. רוב תוכנות הציור (כגון Paint Shop Pro של JASC) מאפשרות לצמצם את גודל לוח הצבעים של התמונה. אם אינך משתמש בצבעים הנוספים, הקטנת עומק הצבע יכולה לחסוך לך מאות בתים בנפח התמונה, דבר המתבטא במהירות הטעינה של התמונה.
- בנה את הגרפיקה שלך בשלבים. פורמט JPEG או GIF תומך בגרפיקה הנבנית בשלבים. כאשר גרפיקה נבנית בשלבים, הדפדפן מתחיל להציג את הגרפיקה עוד לפני שקיבל את המידע המושלם עליה. זהו אפקט פסיכולוגי הגורם לתחושה שהגרפיקה נטענת מהר יותר. כדי לבנות בשלבים גרפיקה בפורמט GIF, בחר לשמור את הגרפיקה בפורמט שזור (Interlaced Format). כדי לבנות בשלבים גרפיקה בפורמט JPEG, בחר לשמור את הגרפיקה עם קידוד הדרגתי (Progressive Encoding).
- כלול תמיד את תכונות הגובה והרוחב של הגרפיקה. דפדפן אינו יכול לעצב את הדף לפני שיש לו את כל הנתונים לגבי גודלם של כל המרכיבים הכלולים בו. תוכל להאיץ את תצוגת הדף על ידי הוספה של התכונות WIDTH ו-HEIGHT

בתגית <IMG>. אם תספק לדפדפן את גובהה ורוחבה של הגרפיקה, הוא יוכל ליצור בדף אזור ריק עם מימדים מתאימים, בזמן שהגרפיקה מועברת.

- אחד גרפיקות נפרדות לגרפיקה אחת. דפדפן חייב לבצע בקשה נפרדת עבור כל גרפיקה הכלולה בדף HTML. אם תאחד גרפיקות רבות לגרפיקה בודדת, לא יהיה צורך בביצוע בקשות רבות על ידי הדפדפן (הכוונה במונח גרפיקה אחת היא ל-MIMI-jpeg  אשר מגדיר את סוג המידע שיועבר. GIf זה סוג אחד ו-JPeg זה סוג אחר). בנוסף, אם הגרפיקות יאוחדו, כיוונצם יהיה יעיל יותר.
- כלול תמיד טקסט חלופי עבור גרפיקה. גולשי אינטרנט מסוימים העייפים מטעינה ארוכה וממושכת של גרפיקה, מחליטים לבטל תצוגת גרפיקה בדפדפן שלהם. מוזר מזה, עדיין ישנם גולשים המשתמשים בדפדפן Lynx שהוא דפדפן מבוסס טקסט בלבד. אם תשתמש בתכונה ALT בתגית <IMG> של HTML כדי לספק טקסט חלופי עבור הגרפיקה שלך, משתמשים אלה עדיין יוכלו להבין את מה שרוצה האתר שלך להביע.

# הימנעות ממסגרות

מסגרות מקילות מאוד על חיי מפתחי האתרים. באמצעות שימוש במסגרות ניתן לחלק בקלות דף אינטרנט לאזורי תוכן נפרדים. למשל, תוכל להשתמש במסגרת אחת כדי לכלול דף ניווט לאתר, ובמסגרת שנייה תוכל לכלול את התוכן המעשי של הדף עצמו. למרות שמסגרות מאפשרות עיצוב פשוט וקל של דף, יש להן בנוסף השפעה המנוגדת לביצועי אתר האינטרנט. הבעיה במסגרות היא שהדפדפן חייב לבצע בקשה נפרדת עבור כל דף במסגרת.מילוי כל הבקשות הנפרדות האלו מתבטא בעומס גדול יותר על השרת.

כמעט בכל המצבים, ניתן להימנע משימוש במסגרות באמצעות שימוש בטבלאות HTML. לדוגמה, אם ברצונך לכלול אזור ניווט משותף לכל דפי האתר שלך, תוכל להשתמש בהפניית INCLUDE# של ASP, כדי להוסיף את התוכן לתא בטבלה בכל עמוד. מנקודת ראותו של המשתמש, ייראה הדף בדיוק אותו דבר כפי שהיה נראה בשימוש חלוקה למסגרות נפרדות.

# שימוש יעיל בטבלאות

במצבים רבים תצטרך להציג רשימת רשומות מטבלת מסד נתונים. לדוגמה, אולי תרצה להציג רשימת אתרי אינטרנט מועדפים, רשימת מוצרים למכירה או רשימת הודעות בקבוצת דיון. כאשר מציגים רשימת רשומות, מפתה להציג את הרשימה בטבלה בודדת כמו בדוגמה הבאה:

```
<TABLE WIDTH="100%">
<% while not RS.Eof %>
<TR>
    <TD><%=RS( "website_name" )%></TD>
    <TD><%=RS( "website_address" )%></TD>
</TR>
```

```
<%
RS.MoveNext
wend
%>
</TABLE>
```

קטע קוד ASP זה מציג ערכת רשומות (RecordSet) על ידי תצוגת שורה נפרדת עבור כל רשומה. אולם, אם ערכת הרשומות מכילה רשומות רבות, ייתכן ודף התוצאות יוצג באיטיות רבה. הבעיה היא שהדפדפן יחכה עד לאחר השלמת הטבלה כולה לפני שיציג חלק כלשהו ממנה. במילים אחרות, הרשומה הראשונה לא תוצג עד לאחר שהרשומה האחרונה תועבר לדפדפן ותישלח התגית המסיימת </TABLE>.

דרך אחת לעקוף בעיה זו היא להשתמש בטבלאות נפרדות כדי להציג כל שורה בערכת הרשומות. הדפדפן יציג את הטבלאות הנפרדות בצורה מיידית. לכן, גם אם אתה מציג ערכת רשומות גדולה מאוד, המשתמש יוכל לראות כל רשומה ברגע שהיא תתקבל. לפניך תסריט משוכתב המציג רשומות בטבלאות נפרדות:

```
<% while not RS.Eof %>
<TABLE WIDTH="100%" BORDER=0>
<TR>
    <TD><%=RS( "website_name" )%></TD>
    <TD><%=RS( "website_address" )%></TD>
</TR>
</TABLE>
<%
RS.MoveNext
wend
%>
```

אם תקבע אותו רוחב לכל העמודות בטבלה ותסתיר את גבול הטבלאות על ידי שימוש ב-BORDER=0, הטבלאות הנפרדות ייראו כטבלה גדולה אחת והדף יוצג בצורה מהירה יותר.

**הערה:**

נקודה חשובה שיש לשים אליה לב כאשר מדובר באיחזור מידע ממסד נתונים אל דף HTML היא שיש לשים לב שהדרך המהירה ביותר היא להביא את כל המידע בפעם אחת (במידה ומסד הנתונים הוא לא מירשם האוכלוסין של מדינת ישראל) ואחר כך להציגו באמצעות שכבות (CSS) במקומות הרצויים, דבר שחוסך זמן ועומס על האתר.

# אופטימיזציה של ביצועי מסד נתונים

אם אתה יוצר אתר מונחה מסד נתונים, עליך להקדיש זמן משמעותי לניטור וכוונון מסד הנתונים, ולבדיקת יעילות הצהרות ה-SQL שלך. אפילו תיקונים קלים יכולים להשפיע בצורה דרמטית על מהירות האתר. בסעיפים הבאים, תלמד מספר שיטות כדי לשפר את מהירות הגישה למסד הנתונים שלך.

**הערה:**

נקודת ההנחה העומדת ביסודו של פרק זה היא שאתה משתמש ב- Microsoft SQL Server (גירסה 6.5 או 7.0). אם ברצונך ליצור אתר שגודלו ניתן לשינוי אל תחשוב אפילו להשתמש ב- Microsoft Access. Microsoft Access היא תוכנת מסד נתונים שולחני (Desktop Database) ולא מסד נתונים מסוג שרת/לקוח (Client/Server Database). משמעות הדבר היא, שהיא לא מתוכננת לתמוך בכמות רבה של משתמשים בו-זמניים אשר אפילו אתר אינטרנט ציבורי צנוע מקבל. לדוגמה, Microsoft Access אינה תומכת בנעילה רבת משתמשים מתקדמת (Advanced Multiuser Locking), ביומני תנועות (Transaction Logs) או במפתוח (Indexing) יעיל.

# מדידת ביצועי מסד הנתונים

כדי לכוונן ביעילות את מסד הנתונים שלך, חייבת להיות לך שיטה בה תוכל למדוד את ביצועיו הנוכחיים. Microsoft SQL Server כולל מספר תוכניות שירות לניטור הפעילות במסד הנתונים ולמדידת המהירות בה מתבצעות השאילתות על מסד הנתונים.

## קביעת זמן ביצוע שאילתות SQL

כדי למדוד את זמן ביצוען של שאילתות SQL, תוכל להשתמש ב-Microsoft ISQL/w (במקרה של Microsoft SQL Server 6.5) או ב-Microsoft Query Analyzer (במקרה של Microsoft SQL Server 7.0). תוכניות שירות אלו תצגנה את כל השלבים אותם מבצע SQL Server כדי להפעיל את השאילתה, ואת הזמן שנדרש לביצוע כל שלב.

כדי לקבוע את משך הביצוע של הצהרת SQL באמצעות Query Analyzer, הפעל את התוכנה מקבוצת תוכניות השירות של SQL Server ובחר ב-Query, Current Connection Options. כעת, בחר באפשרות Show State Time. לאחר שנבחרה אפשרות זאת, בכל פעם שתפעיל שאילתה תוך שימוש ב-Query Analyzer יוצגו, בנוסף לתוצאות השאילתה, גם נתונים סטטיסטיים עבורה.

כדי לראות נתונים סטטיסטיים עבור שאילתה באמצעות שימוש ב-ISQL/w עם Microsoft SQL Server 6.5, הפעל את ISQL/w מקבוצת תוכניות השירות של SQL Server ובחר ב-Query, Set Options. כעת, בחר באפשרות Show State Time. כל אחת מתוכניות שירות אלו תציג את אותם הנתונים. לדוגמה, לאחר ביצוע ההצהרה select count (*) from orders של SQL ב-Query Analyzer, יוצגו הנתונים הסטטיסטיים והתוצאות הבאות:

```
SQL Server Execution Times:
   CPU Time = 0 ms,   elapsed time = 0 ms.
SQL Server parse and compile time:
   CPU Time = 0 ms,   elapsed time = 0 ms.
```

- - - - - - - - - -

```
(1 row(s) affected)

SQL Server Execution Times:
   CPU Time = 0 ms,   elapsed time = 6 ms.

SQL Server Execution Times:
   CPU Time = 10 ms,   elapsed time = 8 ms.
SQL Server parse and compile time:
   CPU Time = 0 ms,   elapsed time = 0 ms.

SQL Server Execution Times:
   CPU Time = 0 ms,   elapsed time = 0 ms.
```

זמן הביצוע של השלבים השונים מוצג במילי-שניות. בדוגמה זו, תוכל לראות שזמן הניתוח וההידור של ההצהרה על ידי SQL Server נמשך פחות ממילי-שנייה, ובערך 8 מילי-שניות כדי לבצע את השאילתה בשלמותה.

## צפייה בתכנית ביצוע השאילתה

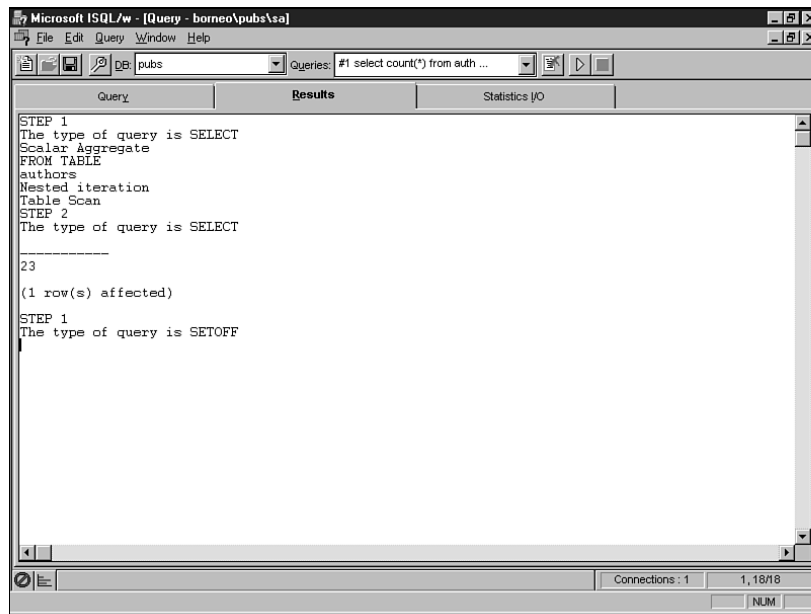
בדרך כלל, ידיעת זמן ביצוע שאילתה תספק לך מידע מספיק כדי לשכתב את השאילתה כך שתפעל באופן יעיל יותר. פשוט המשך לבצע ניסויים עד שהשאילתה תתבצע מהר, אבל ממש מהר. אם בכל אופן יש צורך, גם ISQL/w וגם Query Analyzer יכולות לספק לך ניתוח של התוכנית אחריה עוקב SQL Server כדי לבצע את השאילתה שלך, צעד-אחר-צעד, כולל האינדקסים בהם נעשה שימוש.

כדי לראות את תכנית הביצוע עבור שאילתה באמצעות שימוש ב-ISQL/w בחר ב-Query, Set Options. כעת, בחר באפשרות Show Query Plan. לאחר שבחרת באפשרות זו, ISQL/w יציג את תוכנית הביצוע עבור שאילתה בכל פעם בו הוא יציג את תוצאות השאילתה (ראה תרשים 25.1).

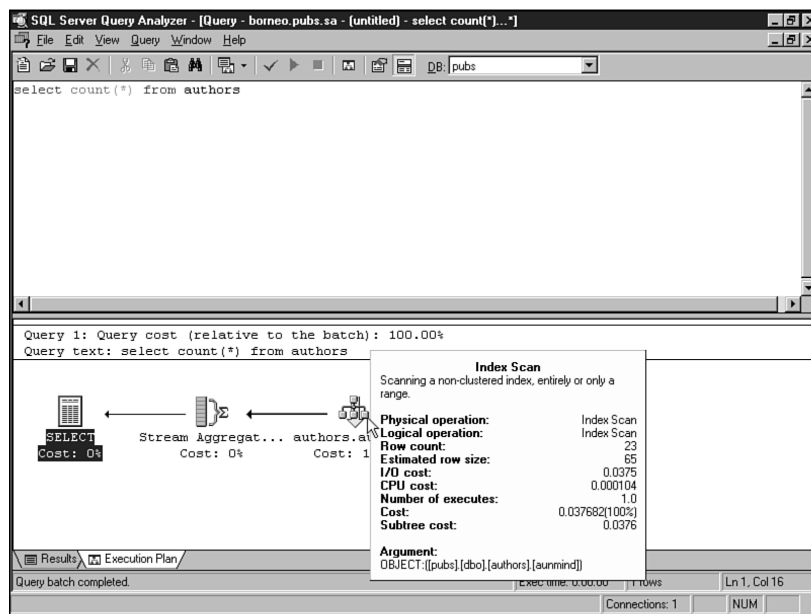
כדי לראות את תכנית הביצוע עבור שאילתה באמצעות שימוש ב-Query Analyzer, בחר ב-Query, Show Execution Plan. לאחר ביצוע שאילתה, תוכל לראות את השלבים אותם ביצע SQL Server כדי לבצע את השאילתה. הצעדים יופיעו בכרטיסיה Execution Plan שבתחתית המסך (ראה תרשים 25.2). כדי לראות נתונים מפורטים לגבי כל אחד מהשלבים, השהה את סמן העכבר על הסמל המייצג את השלב.

## שימוש ב-Profiler

Microsoft SQL 7.0 כולל תוכנית שירות שימושית מאוד בשם **Profiler**. Profiler מאפשר ליצור עקבות (Traces) כדי לנטר את השאילתות הממשיות המבוצעות בשרת מסד הנתונים. לדוגמה, תוכל ליצור עקבות כדי לזהות את השאילתות האיטיות ביותר, או את השאילתות הגורמות למסד הנתונים שלך להיתקע. ניתן להציג עקבות בזמן אמת, או שניתן לרשום אותן לקובץ או לטבלת מסד נתונים לשם אבחון במועד מאוחר יותר.




**תרשים 25.1** תוכנית שאילתה המוצגת באמצעות ISQL/w.

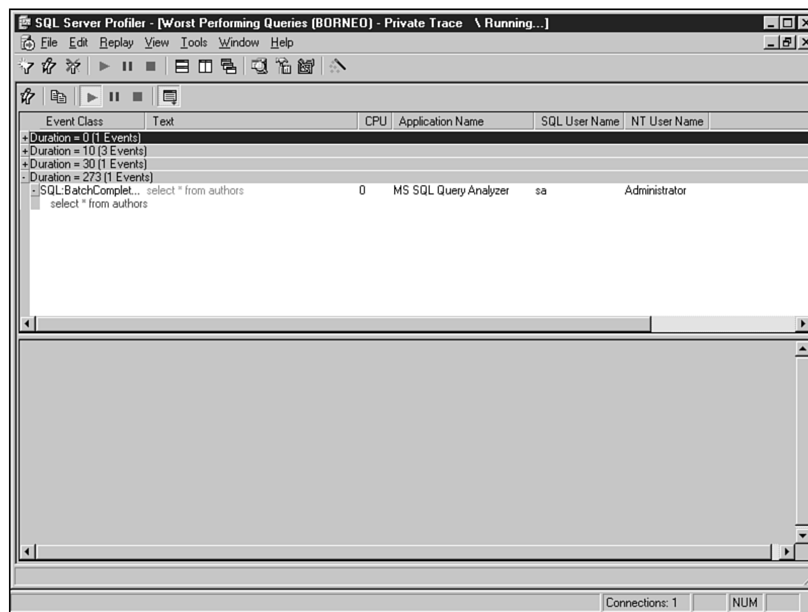



**תרשים 25.2** תוכנית שאילתה המוצגת באמצעות Query Analyzer.

לפניך שלבי השימוש ב-Profiler, בהם תוכל להשתמש כדי לזהות את השאילתות המבוצעות באיטיות הרבה ביותר בשרת מסד הנתונים שלך.

1. הפעל את Profiler מקבוצת תוכניות השירות של SQL Server ובחר ב-Tools, Create Trace Wizard.
2. בחר את שם שרת מסד הנתונים שלך ובאפשרות Find The Worst Performing Queries.
3. כעת, בחר את שמו של מסד הנתונים אותו אתה מעוניין לנטר וציין את משך הזמן המינימלי לשאילתות גרועות. לדוגמה, אם תרצה לזהות שאילתה כלשהי אשר דורשת יותר משנייה לביצועה, הקלד 1000 millisecond.

לאחר ששמרת את העקבות יופיע המסך המוצג בתרשים 25.3. תרשים זה מונה את השאילתות שפועלות יותר משנייה אחת, ברגע שהדבר מתרחש.




**תרשים 25.3** תוכנית השירות Profiler.

# שימוש באינדקסים

בחירתך באינדקסים עבור טבלת מסד הנתונים שלך היא בעלת ההשפעה הגדולה ביותר על ביצועי השאילתות. כאשר השאילתות מבוצעות, תרצה בדרך כלל לעשות כל דבר אפשרי כדי להימנע מסריקת טבלה (Table Scan). סריקות טבלה הן גרועות, כיון שהן מכריחות את SQL Server להתחיל בשורה הראשונה של הטבלה ולקרוא את כל השורות בטבלה כדי למצוא את השורות המתאימות לקריטריוני החיפוש של השאילתה. לדוגמה, כדי לבצע סריקת טבלה של טבלת מסד נתונים המכילה 1,000,000 שורות, חייב SQL Server לקרוא כל אחת מהשורות - תהליך איטי מאוד.

כאשר SQL Server עושה שימוש באינדקסים, הוא מסוגל למנוע את קריאת השורות אחת אחרי השנייה.

SQL Server תומך בשני סוגי אינדקסים: אינדקסים מקובצים (Clustered) ואינדקסים לא מקובצים (Non-Clustered). כל אחד מסוגי אינדקסים אלה טוב יותר לאחזור סוג נתונים מסוים. אינדקס מקובץ פועל בצורה היעילה ביותר באחזור טווח של ערכים. לדוגמה, אם ברצונך לאחזר רשימה של כל העובדים אשר משכורתם נעה בין 6,000 שקל ל- 9,000 שקל מטבלת מסד נתונים הכוללת עמודת שכר, הוספת אינדקס מקובץ לעמודה זו תהיה פעולה יעילה ביותר. טבלת מסד נתונים יכולה להכיל אך ורק אינדקס מקובץ אחד.

אינדקס לא מקובץ פועל בצורה טובה כאשר לעמודה בטבלה יש אחוז גבוה של ערכים שונים. לדוגמה, אם ברצונך לאחזר במהירות שם משתמש בודד מטבלה המכילה עמודה של שמות העובדים, הוספת אינדקס לא מקובץ לעמודה תהיה בחירה טובה. טבלת מסד נתונים יכולה להכיל מספר לא מוגבל של אינדקסים לא מקובצים, בכל כמות אותה תצטרך.

כיון שלאינדקסים השפעה דרמטית כל כך על הביצועים, עליך לערוך כמה שיותר ניסויים עם אינדקסים שונים בטבלת מסד הנתונים שלך. לאחר הוספת או הסרת אינדקס, השתמש ב-ISQL/w או ב-Query Analyzer כדי לבחון את מהירותן של שאילתות אופייניות מול הטבלה המכילה את האינדקס. עליך לקחת בחשבון כי הוספת אינדקסים רבים מדי לטבלה עלולה להאט פעולות שונות, כגון עדכון והוספת ערכים לטבלה. זאת, כיון שהאינדקסים עבור הטבלה חייבים להתעדכן בכל פעם שמתעדכנים נתוני הטבלה.

## בחירת אינדקסים באמצעות שימוש ב-Query Analyzer וב-Profiler

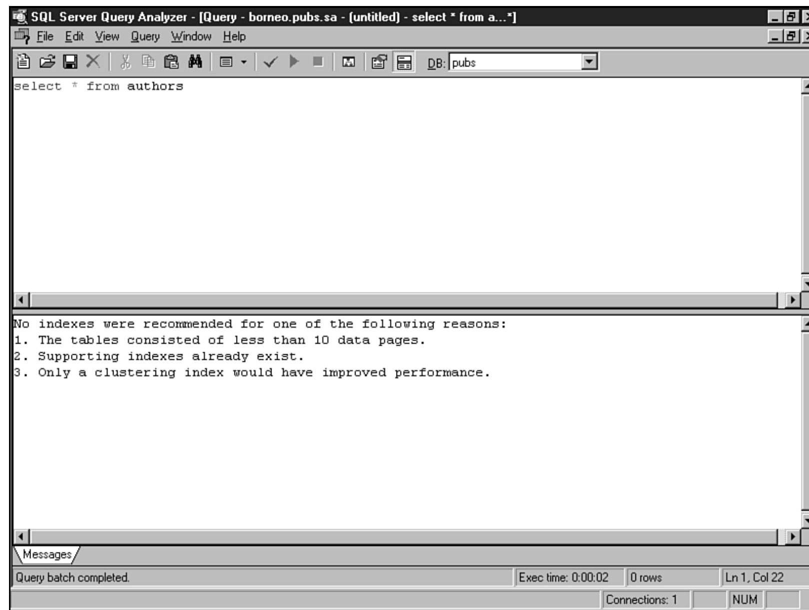
Query Analyzer יכולה לשמש כדי להציע אינדקסים עבור טבלה. כדי להשתמש ב-Query Analyzer על מנת להציע אינדקס, הקלד שאילתת SQL בחלונית העליונה של Query Analyzer ובחר ב-Query, Perform Index Analysis. Query Analyzer תציג את הצעות האינדקס שלה בתחתית חלונית התוצאות (ראה תרשים 25.4).

Query Analyzer תבחן רק את השאילתה אותה הקלדת כדי להציע אינדקס עבור הטבלה. אם מבוצעים מספר סוגי שאילתות שונים מול אותה הטבלה, לא בטוח ש-Query Analyzer תציע את ערכת האינדקסים האופטימלית. יותר מכך, Query Analyzer לעולם אינה מציעה אינדקס מקובץ. כדי לבצע ניתוח אינדקס מדויק יותר, עליך להשתמש ב-Profiler.

Profiler כוללת אשף מיוחד הנקרא Index Tuning Wizard. בעזרת השימוש ב-Index Tuning Wizard תוכל לקבוע את האינדקסים המתאימים ביותר עבור טבלה, כולל אינדקסים מקובצים (Clustered), בהתבסס על עומס העבודה המעשי על SQL Server.

לפני שתוכל להשתמש ב-Index Tuning Wizard, עליך ליצור דבר ראשון עקבות (Traces) כדי לאסוף מידע על סוג השאילתות המבוצעות על שרת מסד הנתונים שלך. כדי ליצור

עקבות אשר ניתן להשתמש בהן עם Index Tuning Wizard, הפעל את Profiler ובחר ב-File, Open, Trace Definition. כעת, בחר בעקבות הנקראים Sample 1 – TSQL ובחר באפשרות Capture to File. לאחר שתשמור את העקבות בכונן הדיסק הקשיח שלך, Profiler יתחיל לשמור נתונים על השאילתות המבוצעות.




**תרשים 25.4** הצעות אינדקס מ-Query Analyzer.

עליך להפעיל את העקבות מספיק זמן כדי ללכוד דוגמה מייצגת של הפעילות במסד הנתונים שלך. לאחר שתחוש מסופק מכמות הנתונים שנאספו, תוכל לעצור את המעקב הפעיל על ידי בחירה ב-File, Stop Traces. כעת, הפעל את Index Tuning Wizard באמצעות בחירה ב-Tools, Index Tuning Wizard.

ב-Index Tuning Wizard תתבקש לספק שם לקובץ העקבות שלך ולבחור את הטבלה אותה ברצונך לכוונן. לאחר שתספק מידע זה, יציע Index Tuning Wizard את הצעותיו לאינדקסים חדשים. תוכל לבחור להרשות לאשף לבצע את שינויי האינדקס בצורה אוטומטית.

## שמירה על עדכניות הנתונים הסטטיסטיים

SQL Server שומר על נתונים סטטיסטיים של התפלגות הערכים השונים בעמודות טבלת מסד הנתונים. הוא משתמש בנתונים הסטטיסטיים הללו כדי לקבוע באלה אינדקסים להשתמש לאחזור מידע מטבלה בזמן ביצוע שאילתה. נתונים סטטיסטיים אלה מתעדכנים בצורה אוטומטית. הם יכולים, בכל אופן, לאבד מרמת הדיוק שלהם אם נתוני הטבלה עוברים סוג כלשהו של שינוי דרמטי. אם נתונים סטטיסטיים עבור טבלה הופכים מיושנים, SQL Server יתחיל לבצע בחירות עלובות לתוכניות ביצוע לשאילתות.

ביכולתך לכפות עדכון של הנתונים הסטטיסטיים בכל טבלאות מסד הנתונים, בטבלה בודדת, או באינדקס בודד עבור טבלה בכל זמן שהוא. כדי לעדכן את הנתונים הסטטיסטיים עבור כל הטבלאות המוגדרות על ידי המשתמש במסד הנתונים, הפעל את שגרת המערכת המאוחסנת sp_updatestats. כדי לעדכן את כל האינדקסים עבור טבלה בודדת, הפעל את ההצהרה UPDATE STATISTICS עם שם הטבלה. לבסוף, ההצהרה הבאה יכולה לשמש לעדכון הנתונים הסטטיסטיים רק עבור אינדקס בשם OrderDate בטבלה הנקראת Orders:

```
Update Statistics Orders OrderDate
```

במקום לעדכן את הנתונים הסטטיסטיים באופן ידני, תוכל להשתמש ב-Database Maintenance Plan Wizard כדי לעדכן את הנתונים הסטטיסטיים עבור טבלה בצורה אוטומטית על בסיס מחזורי. כדי להפעיל את Database Maintenance Plan Wizard, בחר ב-Tools, Maintenance Plan מתוך SQL Enterprise Manager.

## שימוש בשגרות מאוחסנות

ביצוע שגרות SQL מאוחסנות (Stored Procedures) מתוך תסריטי ASP הוא כמעט תמיד שיטה יעילה יותר מאשר ביצוע של מחרוזות שאילתות SQL מקבילות. אם ברצונך להאיץ את ביצועי הגישה למסד נתוני אתר האינטרנט שלך, החלף את כל מחרוזות השאילתות בקריאה לשגרות מאוחסנות.

שגרה מאוחסנת מתבצעת במהירות גדולה יותר ממחרוזת SQL כיון שהיא מנותחת ומהודרת פעם אחת בלבד, בעת ביצועה בפעם הראשונה. בניגוד לכך, בכל פעם בו מבוצעת מחרוזת שאילתת SQL, SQL Server חייב ליצור תוכנית שאילתה חדשה. כדי ללמוד כיצד להשתמש בשגרות מאוחסנות של SQL בתסריטי ASP, ראה פרק 14.

## בחירת סוגי סמן וסוגי נעילה

כאשר אתה מאחזר רשומות מטבלת מסד נתונים, היה בטוח שבחרת את סוגי הסמן וסוגי הנעילה המתאימים. אם לא תציין סוג סמן מסוים בעת פתיחת ערכת רשומות, SQL Server יחזיר את הרשומות באמצעות שימוש במצביע Default Result Set. Default Result Set הוא סוג סמן ספציפי ויעיל מאוד של SQL Server. כאשר נעשה שימוש ב-Default Result Set, נשלחים מייד כל הנתונים משרת מסד הנתונים ללא כל בקשות נוספות מצד הלקוח.

כמעט ואין אף פעם סיבה מחייבת להשתמש ב-dynamic cursor בדפי ASP. במיוחד, יש להימנע בכל מחיר משימוש בסמנים דינמיים (Dynamic Cursors). סמנים דינמיים מפגינים את הביצועים הגרועים ביותר מכל סוגי סמן אחרים. למידע נוסף על בחירת סמנים, ראה פרק 13.

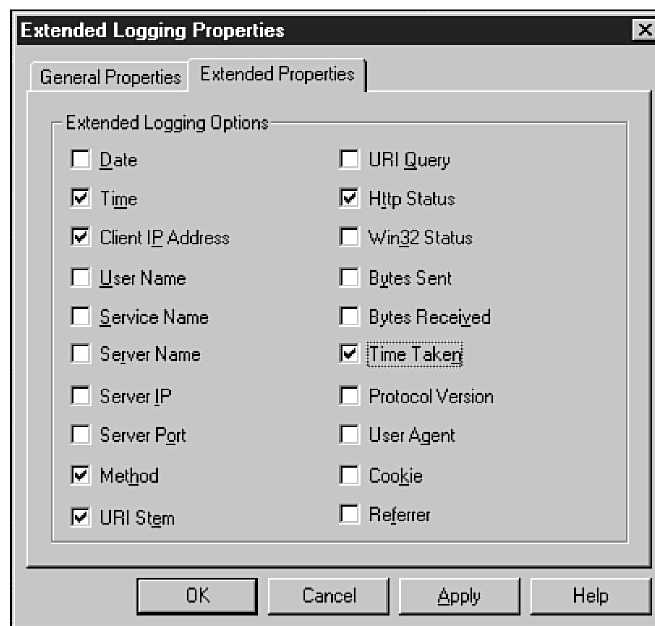
# אופטימיזציה של דפי ASP

בעזרת שימוש במספר טריקים פשוטים, תוכל להגדיל בצורה דרמטית את ביצועם של דפי ה-ASP שלך. בסעיפים הבאים, תלמד מספר שיטות רבות ערך למדידת ביצועי דפי ASP ולכוונון דפי ה-ASP כך שיפעלו בצורה מהירה יותר על ידי שימוש במשאבי השרת באופן יעיל יותר.

## מדידת ביצועי דפי ASP

תוכל לנקוט בשתי גישות כדי למדוד את מהירות הביצוע של דפי ASP. אם אתה מעוניין רק בכמות הזמן הנדרשת לשרת האינטרנט שלך לעבד דף ASP שלם, תוכל להשתמש במאפיינים Extended Logging של Internet Information Server. כדי למדוד חלקים שונים של תסריטי ASP, תוכל להשתמש ב-JScript כדי ליצור תסריט קוצב-זמן מיוחד.

ניתן לתעד את הזמן הכולל הדרוש לעיבוד דף ASP באמצעות שימוש ב-W3C Extended Logging ולאפשר את המאפיין המורחב Time Taken. כדי לעשות זאת, הפעל את Internet Service Manager ופתח את דף המאפיינים עבור אתר האינטרנט שלך. כעת, בחר בכרטיסיה Web Site, ודא שאפשרות יצירת היומן (Logging) מאופשרת וש-W3C Extended Log File Format נבחר. בשלב הבא, לחץ על הלחצן Properties ובחר בכרטיסיה Extended Properties. לכל הפחות המאפיינים URI Stem ו-Time Taken צריכים להיות מאופשרים (ראה תרשים 25.5).

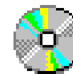



**תרשים 25.5** אפשור המאפיין המורחב Time Taken.

**:הערה**

קיים באג בגירסה הנוכחית של IIS 4.0 הגורם למאפיין Time Taken לרשום ערך שגוי. לפני השימוש במאפיין המורחב Time Taken, ודא שהתקנת את חבילות השירות (Service packs) העדכניות ביותר. למידע נוסף ראה Microsoft Knowledge Base Articles Q191256 ו-Q197817.

לאחר שאפשרת את המאפיין Time Taken, תוכל למדוד את ביצועי דפי ה-ASP שלך על ידי בדיקת קבצי התיעוד (Log files) של האתר. המאפיין Time Taken נרשם בקובץ התיעוד כמספר מסוים במילי-שניות.



אם עליך לקבוע את הזמן עבור חלקים שונים של תסריט בתוך דף ASP, תוכל להשתמש באובייקט JScript Data. כדי למדוד זמן עליך להשתמש בשפת התסריטים JScript ולא ב-VBScript, כיון ש-VBScript אינה כוללת אובייקטים או פונקציות ברמת דיוק גדולה יותר מעל שנייה בודדת. בכל אופן, זוהי אינה מגבלה, כיון שבאפשרותך לשלב בחופשיות תסריט JScript בתוך תסריט VBScript. תדפיס 25.1 ממחיש כיצד תוכל להוסיף פונקציית קוצב-זמן לדף ASP (נמצא בתקליטור תחת השם timer.asp).

**תדפיס 25.1** הוספת קוצב זמן לדף ASP

```
<SCRIPT LANGUAGE="JScript" RUNAT="server">
// timer Function
startTime = new Date();
function showTime()
{
    thisTime = new Date();
    Response.Write( "<HR> elapsed: " )
    Response.Write(  thisTime.getTime() - startTime.getTime() )
    Response.Write( " ms<HR>" )

}
</SCRIPT>

<HTML>
<HEAD><TITLE>Timer</TITLE></HEAD>
<BODY>
<%
for i = 0 to 100
    Response.Write i & " of 100 <BR>"
    showTime()
next
%>
</BODY>
</HTML>
```

תסריט זה עושה שימוש באובייקט JScript Data כדי להחזיר את השעה הנוכחית. בכל פעם בו נקראת הפונקציה ()showTime של JScript, מוצג הזמן שחלף מאז טעינת דף ה-ASP. הפונקציה ()getTime של JScript מחזירה מספר במילי-שניות המציין את הזמן שחלף מאז ה-1 בינואר, 1970, בחצות הלילה (הולדתה של שפת JavaScript Universe). שים לב שהפונקציה ()showTime נקראת באמצע של תסריט VBScript.

# שימוש במשתני Application למניעת גישה למסד הנתונים

דמיין לעצמך שאתה מבצע רכישות דרך אתר עם רשימת מוצרים למכירה. כמו כן, דמיין לעצמך שאינך משנה את רשימת המוצרים לעיתים תכופות, אולי פעם ביום או פעם בכמה ימים. אולי תתפתה לשמור את רשימת המוצרים בטבלת מסד נתונים ולאחזר אותה מטבלה זו בכל פעם שלקוח יבקר באתר האינטרנט שלך. לדוגמה, תוכל להשתמש בקוד ה-ASP הבא כדי להציג את כל המוצרים שלך:

```
<HTML>
<HEAD> <TITLE>My Product</TITLE></HEAD>
<BODY>
<%
Set con = Server.CreateObject( "ADODB.Connection" )

Con.Open "FILE NAME=c:\myDataLink.UDL"
mySQL = "select product_name, product_desc from products"
Set RS = Con.Execute( mySQL )
while not RS.Eof
    %>
    <%=RS( "product_name" ) %> - <%=RS( "product_desc" )%> <P>
    <%
RS.MoveNext
wend
%>
</BODY>
</HTML>
```

דף זה מאחזר את שמותיהם ותיאורם של כל המוצרים מטבלת מסד נתונים הנקראת products. כל אחת מרשומות הטבלה products תוצג בחלון הדפדפן שלך.

אם רשימת המוצרים אינה משתנה לעיתים תכופות, שימוש בדף ASP זה לא תהיה שיטה יעילה במיוחד להצגת הנתונים. גישה למסד נתונים היא תמיד פעולה יקרה במונחים של משאבי שרת. בדוגמה הקודמת, אותו מידע בדיוק מאוחזר מטבלת מסד הנתונים, פעם אחרי פעם.

כיון שהמידע לגבי המוצרים אינו משתנה לעיתים תכופות, רעיון טוב יותר יהיה לאחזר את רשימת המוצרים רק פעם אחת ולשמור אותה לתוך מערך בזיכרון. כדי לעשות זאת, ניתן להוסיף את התסריט הבא לאירוע Application_OnStart של הקובץ Global.asa:

```
SUB Application_OnStart
    Set con = Server.CreateObject( "ADODB.Connection" )

Con.Open "FILE NAME=c:\myDataLink.UDL"
    mySQL = "select product_name, product_desc from products"
    Set RS = Con.Execute( mySQL )
    Application.Lock
    Application( "products" ) = RS.GetRows()
    Application.UnLock
    RS.Close
    Set RS = Nothing
    Set Con = Nothing
End Sub
```

התסריט טוען את רשימת המוצרים מטבלת מסד הנתונים ומשייך את הרשימה למערך משתני Application בשם products. כיון שהתסריט כלול בתת השגרה Application_OnStart, בכל פעם בו מתחיל שירות האינטרנט, רשימת המוצרים נמשכת מהטבלה בצורה אוטומטית ומקושרת למערך זה.

לאחר שרשימת המוצרים מאוחסנת בתוך משתנה בזיכרון, ניתן להציג אותה בדף בלי לאחזר אותה שוב ממסד הנתונים. לדוגמה, התסריט הבא מציג את רשימת המוצרים ממערך Application בשם products:

```
<HTML>
<HEAD> <TITLE>My Product</TITLE></HEAD>
<BODY>
<%
for i = 0 to UBOUND( Application( "products" ), 2 )
    %>
    <%=Application( "products" )(0, i) %> -
    <%=Application( "products" )(1, i) %> </P>
    <%
next
%>
</BODY>
</HTML>
```

שיטת אחזור זו של רשימת המוצרים מהזיכרון יעילה הרבה יותר מאשר אחזור של אותה רשימה מטבלת מסד נתונים פעם אחר פעם.

# הימנעות ממשתני Session

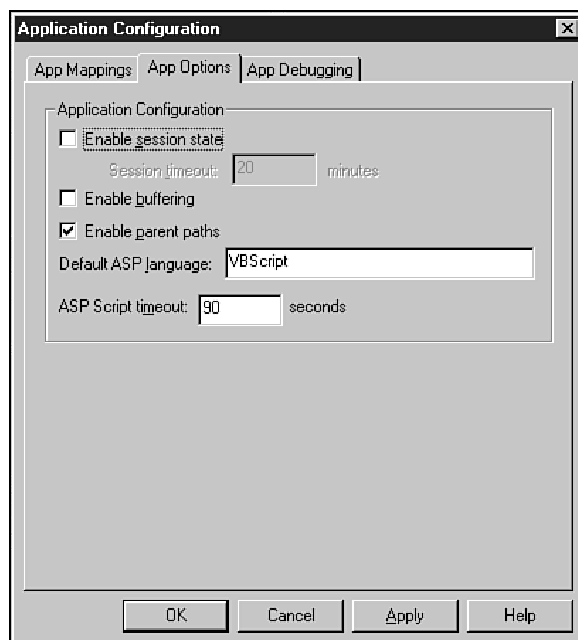
למרות ששימוש במשתני Application יכול להפוך אתר אינטרנט ליעיל יותר, שימוש במשתני Session מוריד כמעט תמיד את יעילות האתר. הבעיה עם משתני Session היא שנוצר מופע חדש של משתנה Session עבור כל מבקר חדש באתר. חמור מכך, בדרך כלל הזיכרון המוחזק על ידי משתנה Session לא משתחרר עד 20 דקות לפחות לאחר שהמשתמש עזב את האתר.

סיבות משכנעות לשימוש במשתני Session הן נדירות מאוד. כמעט בכל המקרים, ניתן להחליף משתני Session בטופס או במשתני מחרוזת שאילתה. או, בתסריט הגרוע ביותר, תוכל תמיד להשתמש ב-cookie. כדי ללמוד עוד על הימנעות משימוש במשתני Session, ראה פרק 5.

אם אינך מתכנן להשתמש במשתני Session בדפי ה-ASP שלך, עליך לבטל אותם באתר. תוכל לעשות זאת בכל דף בנפרד, תוך שימוש בהפניה EnableSessionState של ASP. אם תכלול את השורה הבאה בתחילת הדף, ינוטרלו משתני Session עבור הדף:

```
<% EnableSessionState = FALSE %>
```

כמו כן, תוכל גם להעדיף לנטרל משתני Session לכל האתר שלך באמצעות שימוש ב-Internet Service Manager. כדי לעשות זאת, הפעל את Internet Service Manager, פתח את דף המאפיינים עבור אתר האינטרנט שלך, בחר בכרטיסיה Home Directory, ולחץ על הלחצן Configuration בחלק הנקרא Application Settings. כעת, בחר בכרטיסיה App Options ונטרל את האפשרות הנקראת Enable Session State (ראה תרשים 25.6).

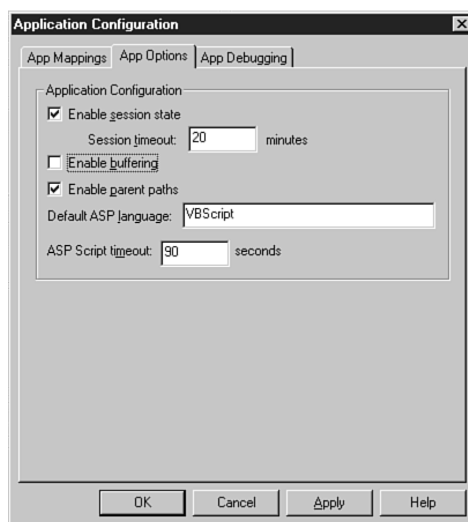



**תרשים 25.6** כיצד לנטרל מצב Session

## אל תאגור את דפי ASP שלך

אגירת דפי ASP באמצעות שימוש במאפיין Buffer של האובייקט Response, הוא בדרך כלל רעיון גרוע. כאשר דף ASP נאגר, שום חלק ממנו אינו מוצג עד שהוא נבנה כולו. במיוחד במקרה של דפי HTML גדולים, דף אגור יכול ליצור השהיה ניכרת.

כברירת מחדל, אגירת דפים מנוטרלת עבור האתר שלך. באפשרותך לאפשר או לנטרל אגירת דפים עבור אתר שלם באמצעות שימוש ב-Internet Service Manager. כדי לעשות זאת, הפעל את Internet Service Manager, פתח את דף המאפיינים עבור אתר האינטרנט שלך, בחר בכרטיסיה Home Directory, ולחץ על הלחצן Configuration בחלק הנקרא Application Settings. כעת, בחר בכרטיסיה App Options ואפשר או נטרל את האפשרות Enable Buffering (ראה תרשים 25.7).




**תרשים 25.7** כיצד לעצב אגירת דפים. שים לב שב-IIS 5 האגירה היא ברירת מחדל ואילו ב-IIS 4 לא.

# תכנון קיבולת

בחלק אחרון זה של הפרק, תוצג לפניך סקירה כללית קצרה על InetMonitor של Microsoft. הכלי InetMonitor תוכנן לאפשר לך לנטר ולבחון את קיבולת אתר האינטרנט שלך באופן מקיף. כלי זה כולל שני רכיבים. באמצעות שימוש ברכיב הניטור (Monitor Component), תוכל ללכוד מידע על התנהגות טיפוסית של מבקרים באתר. באמצעות שימוש ברכיב ההדמייה (Simulation Component), תוכל לבצע הדמיה של מאות או אלפי משתמשים סימולטניים ולבחון את טיב תפקודו של האתר תחת תעבורת משתמשים כבדה.

**הערה:**

InetMonitor 3.0 נכלל ב-Microsoft BackOffice Resource Kit (מהדורה שנייה).
באפשרותך לטעון אותו ישירות מהאתר של Microsoft בכתובת:
http://www.microsoft.com/msdownload.

**הערה:**

ל- Microsoft יש כלי ניתוח קיבולת נוסף בשם Web Capacity Analysis Tool (WCAT). ניתן לטעון את WCAT ישירות מהאתר של Microsoft (בצע חיפוש עבור "WCAT" בכתובת http://www.microsoft.com/search) או לרכוש אותו יחד עם Internet Information Server Resource Kit.

בהתקנת InetMonitor מעורבים שני טריקים. ראשית, לאחר התקנת התוכנה אתה חייב למצוא את התיעוד שלה. התיעוד חבוי במסמך Microsoft Word באותה הספריה בה הותקנה התוכנה (כברירת מחדל, c:\InetMonitor).

כמו כן, לפני שתוכל להשתמש ב-InetMonitor לביצוע מבחני קיבולת לאתר, עליך לכייל את כונן הדיסק הקשיח שלך. תוכל לדלג על שלב זה במשך זמן היכרותך את התוכנה ולבצע את הכיול בזמן מאוחר יותר. עיין בעזרה המקוונת ל-InetMonitor או במסמך ה-Word כדי ללמוד כיצד לבצע את הכיול.

אחד מיתרונותיו הגדולים ביותר של InetMonitor הוא קלות השימוש בו. כדי להתחיל ולנטר את האתר שלך בחר ב-Monitoring, Monitor Server והקלד את שם השרת שלך. ברגע שתבחר את השרת, כל השירותים הנתמכים על ידו - לדוגמה IIS, SMTP, או NNTP - יופיעו בחלון השמאלי שעל המסך. כדי לראות את המאפיינים של שרת האינטרנט שלך, לחץ על השירות IIS. לאחר שתעשה זאת, תוכל לנטר מספר היבטים של ביצועי השרת באמצעות בחירה בכרטיסיות שונות.

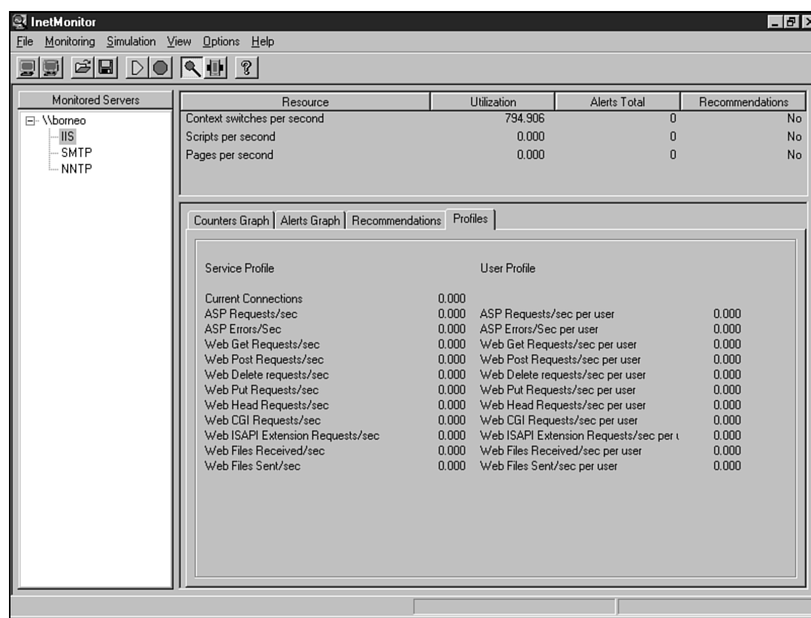
לדוגמה, בכרטיסיה Profiles, תוכל לראות את פרופיל שירות האינטרנט שלך ופרופיל של משתמש אופייני באתר (ראה תרשים 25.8). InetMonitor עוקב אחרי מידע כזה באמצעות מספר בקשות ה-ASP שמבצע משתמש אופייני בשנייה ובאמצעות מספר השגיאות שמשתמש אופייני כזה מקבל.

היישום המעניין ביותר של InetMonitor משמש לביצוע הדמיית עומסים על שרת האינטרנט שלך. InetMonitor הוא כלי גמיש מאוד. תוכל ליצור תסריטים מותאמים לביצוע הדמיה עבור כל סוג של פעילות משתמש. שפת התסריטים תומכת ב-cookies, טפסים שונים של אימות משתמש, ובשני הפרוטוקולים HTTP 1.0 ו- HTTP 1.1.

כדי ליצור הדמיה, עליך ליצור תחילה תסריט המכיל רשימת פעולות עבור המשתמש המדומה. לדוגמה, נניח שהאתר שלך כולל שלושה דפים הנקראים test1.asp, test2.asp ו-test3.asp, כולם ממוקמים בספריית השורש. תסריט הפקודה הבא מבצע בקשה לכל אחד מדפים אלה, לפי הסדר:

```
GET url:/test1.asp
GET url:/test2.asp
GET url:/test3.asp
```

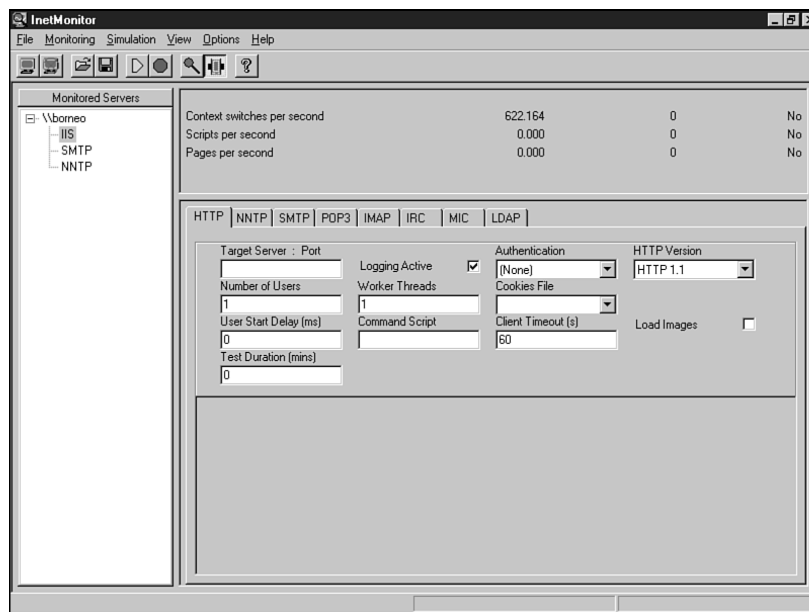
הקלד תסריט פשוט זה לתוך פנקס הרשימות (Notepad) ושמור אותו במקום כלשהו בכונן הדיסק הקשיח שלך. כדי להפעיל את ההדמיה, בחר בתצוגת ההדמיה מ-InetMonitor על ידי בחירה ב-View, Simulation View. תראה את המסך המופיע בתרשים 25.9. הקלד את שמו של שרת האינטרנט שלך בתיבה Target Server, את הנתיב לקובץ התסריט ששמרת כרגע בתיבה Command Script, ואת מספר המשתמשים שברצונך לדמות בתיבה Number of Users. כדי להתחיל את ההדמיה, בחר ב-Simulation, Run Simulation.




**תרשים 25.8** התוכנה InetMonitor.

במשך זמן פעולת ההדמיה, תוכל לראות נתונים סטטיסטיים שונים כגון, מספר חיבורי המשתמשים (user connections), דפים שהתקבלו, ושגיאות בזמן אמת (כפי שמוצג בתרשים 25.8). כמו כן תוכל להחליט לתעד את ההדמיה לקובץ טקסט על ידי בחירה ב-Monitoring, Log Active.

לאחר שביצעת הדמיה, תוכל לקבל המלצות מ-InetMonitor לאפשרויות שיפור שונות של ביצועי שרת האינטרנט שלך. לדוגמה, התוכנה תוכל להציע להתקין זיכרון נוסף או מעבד מהיר יותר.




**תרשים 25.9** ביצוע הדמיית תעבורה עם InetMonitor.

# סיכום

בפרק זה למדת מספר שיטות בעלות ערך למדידת הקיבולת של אתר האינטרנט שלך ולשיפור הביצועים שלו. למדת כיצד לבצע אופטימיזציה של דפי HTML, גישה למסד הנתונים ולתסריטי ה-ASP שלך. לבסוף, ערכת הכרת את InetMonitor, כלי שיכול לשמש לביצוע בדיקות מתוחכמות ביותר של קיבולת אתר האינטרנט שלך.

## פרק 26

# Win 2000, IIS 5.0, ASP - כולם ביחד

**בפרק זה:**

- מבט על Windows 2000 ו-IIS 5.0
- מה חדש ב-VBScript 5.0
- הרחבות לאובייקטים המובנים של ASP
- הרחבות לרכיבים המותקנים של ASP

Windows 2000 כוללת מספר תכונות חשובות חדשות עבור מפתח ASP. Windows 2000 כוללת גרסאות חדשות ל-IIS, VBScript, ActiveX Data Objects, והחשוב מכל, היא כוללת גירסה חדשה ל-ASP.

**הערה:**

למידע נוסף על Windows 2000, בקר באתר האינטרנט Windows 2000 של Microsoft, המעדכנת מעת לעת את גרסאות מוצריה וכדי להתעדכן באתר, בכתובת: http://www.microsoft.com/windows2000.

מטרת פרק זה היא לספק סקירה כללית על השיפורים החשובים ביותר ל-IIS, VBScript ו-ASP הכלולים ב-Windows 2000. חלק משינויים אלה ישפיעו ברצינות על הדרך בה תתכנת אתרי Web (שיפור משמעותי ביותר). אפילו אם אינך מתכנן שדרוג מיידי ל-Windows 2000, עליך ללמוד להכיר את התכונות החדשות שלה, כדי שתוכל להתחיל בהכנת אתרי האינטרנט שלך לעתיד.

# מבט על חלונות 2000 ו-IIS 5.0

IIS גירסה 5.0 נראה דומה מאוד ל-IIS 4.0. זאת כיון ש-IIS 4 היווה הצצה ל-Windows 2000 עם רעיון MMC (Microsoft Management Console) בשימוש המעשי הראשון שלו. למרות הרושם, Microsoft שיפרה את IIS בארבעה תחומים חשובים:

- שיפור בביצועים של IIS על ידי היכולת לדחוס דפים בזמן שהם משודרים מהשרת.
- שיפור בתכונות האבטחה של IIS על ידי הוספת פרוטוקול אימות חדש והוספת אשפי אבטחה חדשים.
- ל-IIS יכולת משופרת לזיהוי דפי ASP.
- לבסוף, Microsoft הוסיפה **תכונות חיבור מרחוק** (Remote Authoring Features) חדשות ל-IIS על ידי יישום פרוטוקול Distributed Authoring and Versioning (DAV).

# דחיסת HTTP

ככל שקטנה כמות המידע שצריכה להיות משודרת מהשרת לדפדפן, כך יגיע לשם המידע במהירות גדולה יותר. IIS 5.0 כולל אפשרות לדחיסת דפי אינטרנט לפני שהם נשלחים לדפדפן. תכונה זו עובדת רק עם דפדפנים המאפשרים דחיסה (במילים אחרות, Internet Explorer 5.0 ולא Netscape). דפדפן שאינו יכול לעבוד עם דפים דחוסים עדיין יפעל כשדחיסת HTTP מאופשרת ב-IIS - הוא פשוט לא יקבל קבצים דחוסים.

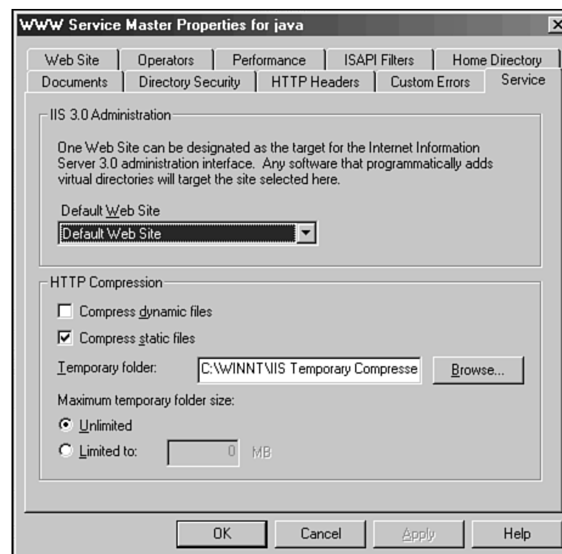
תוכל לבחור לדחוס רק דפי HTML סטטיים, רק דפי ASP, או את שניהם. אם אתה מאפשר דחיסה, אתה חייב לאפשר אותה לכל אתר האינטרנט שלך. אינך יכול לאפשר דחיסה לכמה ספריות בלבד ולא לאחרות.

נראה כיצד דחיסת HTTP פועלת במקרה של דפים סטטיים. כאשר מישהו מבקש דף מאתר האינטרנט שלך, IIS 5.0 בודק האם הדפדפן מקבל קבצים דחוסים. אם הדפדפן אכן מקבל דפים דחוסים, השרת בודק בספריית דחיסה מיוחדת אם הקובץ המבוקש כבר קיים בצורה דחוסה. אם הקובץ כבר קיים, השרת שולח אותו. אחרת, אם זו הפעם הראשונה שהקובץ התבקש, ישלח השרת גירסה לא דחוסה של הקובץ המבוקש וייצור גירסה דחוסה בספריית הדחיסה עבור הפעם הבאה שהדף יתבקש.

דחיסת HTTP פועלת בצורה שונה במקצת במקרה של דפי ASP. כיון שדף ASP עשוי להיות שונה בכל פעם בה הוא נוצר, IIS 5.0 לעולם לא ישמור עותק של דף ASP דחוס בספריית הדחיסה. לחילופין, השרת יבצע דחיסה מחודשת של דף ה-ASP וישלח אותו בכל פעם שהוא יתבקש.

ברור שדחיסת HTTP עובדת הרבה יותר טוב עם קבצי HTML סטטיים מאשר עם דפי ASP. אם אתה מאפשר דחיסה עבור דף ASP, אתה עשוי להכביד על מעבד המחשב, כיון שהמעבד חייב לעבוד כדי לדחוס דף ASP בכל פעם שהדף מבוקש. מסיבה זו, הגיוני לדחוס רק דפי HTML.

כדי להפעיל דחיסת HTTP, הפעל את Internet Services Manager ופתח את כרטיסיית המאפיינים עבור אתר האינטרנט שלך. לאחר בכן, בחר ב-WWW Service ולחץ על הלחצן Edit. לבסוף, בכרטיסיה Service, בחר באפשרויות הדחיסה אותן ברצונך להפעיל (ראה תרשים 26.1).




**תרשים 26.1** הפעלת דחיסת HTTP.

כמה יעילה היא דחיסת HTTP? כדי לקבל תחושה עד כמה מסוגלת דחיסה זו להקטין גודל של קובץ, תוכל להשוות את גודלו המקורי של הקובץ לגודלו של הקובץ הדחוס הממוקם בספריית הדחיסה (כברירת מחדל, ממוקמת ספריית הדחיסה ב-WINNT\IIS-Temporary Compressed Files). לדוגמה, קובץ שמכיל רק 1000 תווי X בשורה, מכיל בדיוק 1000 בתים. לאחר שהקובץ נדחס על ידי IIS 5.0, הקובץ מכיל רק 38 בתים. זהו מקרה קל, בכל אופן, כיון שאלגוריתמים לדחיסה יעילים במיוחד כאשר הם משמשים לדחיסת רצף ארוך של מידע זהה.

כמבחן נוסף, טענתי דף אופייני מהאינטרנט. גודלו של הקובץ המקורי היה 50,573 בתים, וגודלה של הגירסה הדחוסה היה רק 13,048 בתים. במילים אחרות, הגירסה הדחוסה של הדף מהווה רק 26% מהגודל המקורי. לא רע.

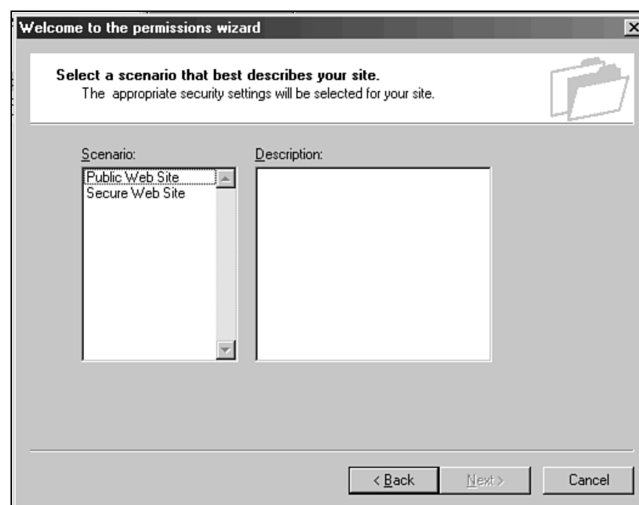
תוכל להשוות בין הגרסאות הדחוסות והגרסאות הלא דחוסות של דפים באתר האינטרנט שלך, כדי להחליט האם עליך להשתמש בדחיסה. בזמן הניסוי זכור להשגיח על הביצועים של מעבד שרת האינטרנט שלך. כדי להשגיח על ביצועי השרת, השתמש ב-System Monitor של Windows 2000 (התחליף ל-Performance Monitor ב-Windows NT).

# אבטחה ב- IIS 5.0

IIS 5.0 כולל שני שיפורי אבטחה משמעותיים. תחילה, נוספה שיטת אימות חדשה בשם Digest Authentication. Digest Authentication (תקציר אימות) דומה ל-Windows NT Challenge/Response Authentication, מלבד העובדה שהוא פועל גם כאשר נעשה שימוש בשרתי Proxy וב-Firewalls. בדומה ל-Windows NT Challenge/Response Authentication, סיסמאות לעולם לא נשלחות ברשת כטקסט פשוט. כמו כן, בדומה ל-Windows NT Challenge/Response Authentication, שיטת אימות זו פועלת רק עם Microsoft Internet Explorer (Windows NT Challenge/Response Authentication פועלת עם Internet Explorer גירסה 2.0 ומעלה, ו-Digest Authentication פועלת רק עם Internet Explorer גירסה 5.0 ומעלה).

IIS 5.0 כולל גם אשפי אבטחה חדשים עבור בקשת והתקנת אישורים (Certificate Wizard), ניהול רשימות אישורי נאמנים - Certificate Trust Lists (CTL Wizard), והגדרת הרשאות לקבצים ולספריות (Permissions Wizard). Permissions Wizard הוא אשף שימושי במיוחד. תוכל להשתמש באשף זה כדי לתאם בין הרשאות קבצי NTFS עבור קבצי דף האינטרנט שלך לבין אבטחת IIS.

אשף ההרשאות כולל שתי תבניות (Templates) עבור הגדרות תצורה של שני סוגי שרתי האינטרנט הנפוצים ביותר. תוך שימוש בתבנית Public Access Internet/Intranet Web Site, תוכל להעניק באופן אוטומטי הרשאות קריאה בלבד לקבצי אתר האינטרנט שלך עבור כל המשתמשים הציבוריים, ולתת למנהלי הרשת שליטה מלאה. תוך שימוש בתבנית Secure Internet Web Site, תוכל להגביל את הכניסה רק למשתמשים מאומתים. כדי להפעיל את אשף ההרשאות, פתח את Internet Services Manager, בחר את אתר האינטרנט שלך, פתח את תפריט Action, בחר Task, Permissions Wizard (ראה תרשים 26.2).




**תרשים 26.2** אשף ההרשאות (Permission Wizard).

## דפי ASP ללא תסריטים

בעת יצירת אתר אינטרנט, אני הופך תמיד את כל הדפים לדפי ASP, על ידי שמירתם עם הסיומת asp. אני עושה זאת גם במקרים של דפי HTML סטטיים, אשר כלל אינם מכילים תסריטי ASP. אני יוצר כל דף כדף ASP כיון שאני יודע שקיימת אפשרות שאולי אאלץ להוסיף תסריטים לדף בעתיד. אני דואג לכך שהמרת דפי HTML לדפי ASP על ידי שינוי הסיומת htm. לסיומת asp. תגרור קישורים שבורים במנועי חיפוש או בדפים אחרים באתר האינטרנט.

התנהגות זו מדאיגה אותי תמיד. IIS 4.0 חייב לעבוד קשה כדי לעבד דף ASP, כיון שהוא חייב לשלוח כל דף עם הסיומת asp. למנוע ASP ‏(asp.dll), ללא התחשבות בעובדה האם הדף כולל תסריטים או לא. לכן, המרה של כל הדפים לדפי ASP עלולה לגרום לירידה מסוימת בביצועי האתר.

IIS 5.0 הקל על דאגתי. IIS 5.0 חכם מספיק כדי לזהות האם דף אכן כולל למעשה תסריטי ASP. אם הדף אינו מכיל כלל תסריטים, הוא נשלח לדפדפן במהירות רבה יותר. לכן, לא קיימת יותר סיבה מניחה את הדעת שלא להפוך את כל הדפים לדפי ASP. ‏IIS 5.0 מבצע זאת על ידי סריקת הדף וחיפוש אחר תגיות <script>.

## Distributed Authoring and Versioning

Distributed Authoring and Versioning ‏(DAV) הוא הרחבה של פרוטוקול HTTP המאפשר למשתמשים לטפל בקבצים ובספריות בשרת אינטרנט ברשת האינטרנט. תוכן להשתמש ב-DAV לביצוע פעולות שונות, כגון העברה, העתקה ומחיקה של קבצים ותיקיות. בנוסף, תומך DAV גם בנעילת קבצים, כך שמשתמשים רבים יכולים לגשת ולבצע שינויים בקבצים המאוחסנים בשרת האינטרנט מבלי לדרוס אחד את השינויים של השני.

DAV מספק תפקודיות כמעט זהה לזו שמספק הפרוטוקול המשמש את Microsoft FrontPage לעריכת דפי אינטרנט ברשת האינטרנט, מלבד העובדה ש-DAV הוא תקן פתוח. המפרט מסופק במסמך Internet Engineering Task Force (IETF) Internet Draft (ראה /http://www.ietf.org). כיון שזהו תקן IETF פתוח, תקן DAV אמור להיות נתמך על פני תחום רחב של לקוחות ושרתי אינטרנט (לא רק Microsoft). ‏Microsoft תומכת כרגע ב-DAV ב- Internet Explorer 5.0 וב- Office 2000.

# מה חדש ב- VBScript 5.0

Windows 2000 כוללת גירסה חדשה של VBScript 5.0. גירסה חדשה זו של VBScript כוללת תכונות חדשות עבור שליטה אוטומטית בביצוע תוכניות, עבודה עם אובייקטים ומחלקות, שימוש ב**ביטויים רגולריים** (Regular Expressions), והסתרת קוד מקור. באפשרותך להשתמש בתכונות חדשות אלו הן בדפי ASP והן בתסריטי Windows) WSH Scripting Host).

# הפעלת קוד באופן דינמי

ל-VBScript נוספה הצהרה חדשה אשר מאפשרת ביצוע של קוד בזמן ריצה. באמצעות שימוש בהצהרה Execute, תוכל לבצע כל מחרוזת המכילה קוד VBScript תיקני.

לדוגמה, התסריט הבא מקשר את כל הפריטים מהאוסף Form של האובייקט Request למשתנים מקומיים של VBScript :

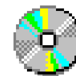
```
<%
for each thing in Request.Form
    Execute thing & "  =" "  " & Request.Form(  thing  ) & "  ""  "
next
%>
```

אם טופס הכולל שדה טקסט בשם Password יועבר לדף ASP הכולל תסריט זה, אזי ערכו של שדה הטופס יקושר באופן אוטומטי למשתנה מקומי של VBScript בשם password.

המחרוזת המועברת להצהרה EXECUTE יכולה לכלול הצהרות VBScript רבות. על כל הצהרה להיות מופרדת באמצעות נקודתיים. לדוגמה, התסריט הבא מקשר שני ערכים למשתנה בשם myvar :

```
<%
theCode = "myvar=3:myvar=67"
Execute theCode
%>
```

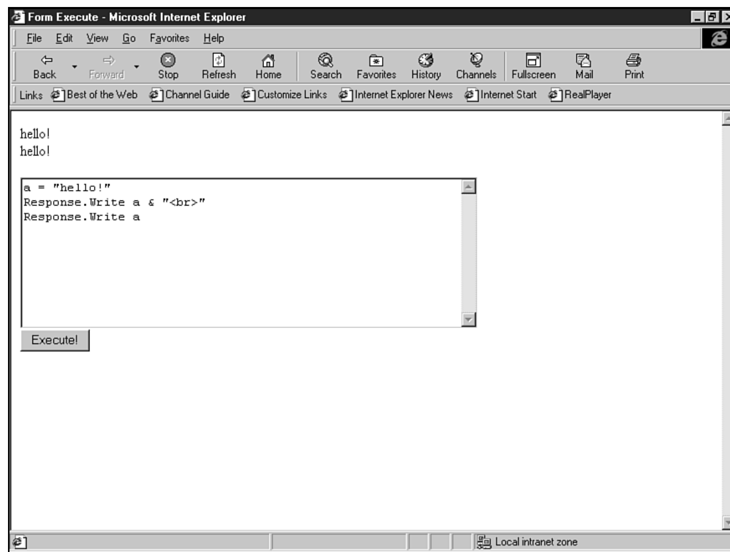
לאחר ביצוע תסריט זה, יכיל המשתנה את הערך *67*.



חשוב להבין שניתן להשתמש בכל קוד VBScript רגיל עם ההצהרה EXECUTE, ובכלל זה כל קוד המתייחס לאובייקטים של ASP. קוד ASP הבא (נכלל בתקליטור בשם formExecute.asp) כולל קטע קוד HTML המאפשר למשתמשים להקליד תסריט. כאשר הטופס נשלח, התסריט מבוצע (ראה להלן תרשים 26.3) :

```
<!-- FormExecute.asp -->
<%
On Error Resume Next
theCode = Request( "theCode" )
theCodeArray = Split( theCode, vbCRLF )
theScript = Join( theCodeArray, ":" )
%>
<html>
<head><title>Form Execute</title></head>
<body>
<% Execute theScript %>
```

```
<form method="post" action="FormExecute.asp">
<textarea name="theCode"
 cols=60 rows=10><%=theCode%></textarea>
<br><input type="submit" value="Execute!">
</form>

</body>
</html>
```




**תרשים 26.3** הטופס לביצוע תסריטים

תסריט זה מאחזר את התסריט שהוקלד לתוך אזור הטקסט ומפצל אותו לתוך מערך באמצעות שימוש בפונקציה ()SPLIT של VBScript. כעת, נעשה שימוש בפונקציה ()JOIN כדי לאחד בחזרה את חלקי התסריט השונים, באמצעות נקודתיים המשמשות כחוצצים בין ההצהרות השונות. לבסוף, התסריט מבוצע באמצעות ההצהרה EXECUTE.

**אזהרה:**

אל תשתמש בתסריט כגון זה באתר אינטרנט ציבורי, כיון שהוא חושף מספר חורי אבטחה חמורים מאוד. לדוגמה, מישהו יכול להחליט להפעיל תסריט המשתמש ברכיב File Access, כדי למחוק את כל הקבצים מכונן הדיסק הקשיח של שרת האינטרנט שלך.

## הערכת קוד באופן דינמי

ההצהרה Execute אשר תוארה בסעיף הקודם, מאפשרת לבצע קוד VBScript בזמן ריצה. במצבים מסוימים, בכל אופן, תרצה אולי רק להעריך ביטוי, ולא לבצע אותו. כדי להעריך ביטוי, תוכל להשתמש בפונקציה ()EVAL של VBScript. לדוגמה, התסריט הבא מציג את הערך TRUE כיון שהערך 1 קטן מהערך 2 :

```
<%
result = EVAL( "1<2" )
Response.Write result
%>
```

## עבודה עם מחלקות

אם אתה חובב של תכנות מונחה-עצמים או מכיר את Visual Basic, בוודאי תשמח לגלות ש- VBScript 5.0 תומכת במחלקות. **מחלקה** (Class) היא אובייקט המוגדר לפי הצרכים האישיים המכיל מאפיינים ושיטות מותאמות אישית. תוכל להשתמש במחלקות כדי לבנות את הקוד שלך בצורה יעילה יותר וכדי לשלוט על האופן בו נקראים ומעודכנים המשתנים.

לדוגמה, בתוך תסריטי ASP, תצטרך אולי לעקוב ולעדכן נתונים על המבקרים באתר האינטרנט שלך. כדי לבצע זאת, תוכל להגדיר מחלקה המייצגת משתמש אינטרנט :

```
<%
Class WebUser
    Privet Username, Password
    Public Defualt Property Get theUsername
        theUsername = username
    End Property
    Public Property Get checkPassword( thePassword )
        if thePassword = Password then
            checkPassword = TRUE
        else
            checkPassword = FALSE
        end if
  End Prperty
  Public SUB addWebuser( theUsername, thePassword )
        Username = theUsername
        Password = thePassword
    END SUB
END CLASS
%>
```

מחלקה זו מגדירה שיטה אחת ושני מאפיינים. השיטה ()addWebuser משמשת לאתחול המשתנים Username ו-Password. המאפיין checkPassword בודק אם הסיסמה נכונה. לבסוף, מאפיין ברירת המחדל של המחלקה מחזיר את שם המשתמש.

כדי להשתמש במחלקה זו בתוך תסריט, עליך ליצור תחילה מופע של המחלקה, באמצעות שימוש בהצהרה NEW. לאחר יצירת מופע של המחלקה, תוכל להתייחס אליה בדיוק באותו אופן בו אתה מתייחס לאובייקטים אחרים של ASP:

```
<%
' Create an instance of the class
Set aUser = new WebUser
' Add a new user
aUser.addWebuser "Andrew Jones", "secret"
' Check the password
Response.Write "Is 'dodgy' the right password?"
Response.Write aUser.CheckPassword( "dodgy" )
' Display the username
Response.Write "<BR>" & aUser
%>
```

שים לב שההצהרה האחרונה בתסריט מחזירה את שם המשתמש, ללא ציון מאפיין. דבר זה פועל כיון שמאפיין ברירת המחדל של המחלקה WebUser הוא שם המשתמש.

מדוע להשתמש במחלקות? יתרונה העיקרי של המחלקה הוא בכך שהיא מספקת שיטה לשליטה באופן בו ניתן לגשת למשתנים ולשנותם. מחלקה יכולה להגן על משתנים משינויים בלתי מכוונים.

בדוגמה הקודמת, המשתנה Password הוצהר כמשתנה מסוג Private. משמעות הדבר היא, שהדרך היחידה בה ניתן לקרוא או לשנות משתנה זה היא באמצעות קריאה לשיטות ולמאפיינים של המחלקה WebUser. כיון שהמחלקה WebUser אינה כוללת מאפיין להחזרת ערך המשתנה Password, המידע נשמר חסוי.

**הערה:**

מידע נוסף על VBScript תוכל למצוא באתר MIcrosoft בכתובת:
. www.microsoft.com/scripting

# עבודה עם אובייקטים

VBScript 5.0 כוללת הצהרה חדשה עבור עבודה עם אובייקטים - ההצהרה WITH. בדרך כלל, כאשר אתה עובד עם אובייקטים, אתה חייב לספק את שם האובייקט בכל פעם שאתה קורא לאחת מהשיטות, או לאחד מהמאפיינים שלו. לדוגמה, נניח שעליך להכניס רשומות רבות לטבלת מסד נתונים, באמצעות האובייקט Connection:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )

Con.Open "FILE NAME=c:\myDataLink.UDL"
Con.Execute "insert ( aNumber ) values ( 1 )"
Con.Execute "insert ( aNumber ) values ( 2 )"
Con.Close
%>
```

בכל פעם שאתה קורא לשיטה Execute, עליך לפנות לאובייקט Connection. לעומת זאת, אם תשתמש בהצהרה WITH, תוכל להצהיר על אובייקט ברירת מחדל, ולהשמיט את שם האובייקט:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
With Con

   .Open "FILE NAME=c:\myDataLink.UDL"
   .Execute "insert ( aNumber ) values ( 1 )"
   .Execute "insert ( aNumber ) values ( 2 )"
   .Close
End With
%>
```

כל קריאה לשיטה או למאפיין אשר מופיעים בין ההצהרות WITH ו-END WITH, משתמשות באובייקט Con. המטרה הבלעדית של ההצהרה WITH היא פשוט לחסוך לך קצת הקלדה.

# ניתוח מחרוזות באמצעות ביטויים רגולריים

אחת הסיבות העיקריות לפופולריות של שפת כתיבת התסריטים PERL היא תמיכתה **בביטויים רגולריים** (Regular Expressions). ביטויים רגולריים יכולים לשמש לביצוען של משימות טיפול מורכבות במחרוזות, כגון **התאמת תבניות** (Pattern matching) **והחלפת מחרוזות** (String replacement). VBScript 5.0 תאפשר לך להתרברב בפני חבריך הכותבים בשפת PERL ש-VBScript גם היא תומכת בביטויים רגולריים. VBScript 5.0 תומכת במבנה בסיסי של התאמת תבניות ביטויים רגולריים עם שני אובייקטים חדשים: האובייקט RegExp והאובייקט Match.

תוכל להשתמש בביטויים רגולריים כדי לבדוק תקפות נתונים שמולאו בטופס HTML. לדוגמה, תוכל להשתמש בביטוי רגולרי כדי לבדוק האם משתמש הקליד כתובת דואר אלקטרוני אמיתית, מספר ביטוח לאומי אמיתי, או מספר טלפון אמיתי, באמצעות התאמת הנתונים שמולאו על ידי המשתמש מול תבנית מחרוזת המוגדרת על ידי ביטוי רגולרי.

ביטוי רגולרי מכיל רצף של תווים מיוחדים המגדירים תבנית. לדוגמה, הביטוי הרגולרי "r[ao]t" מתאים למחרוזת "rat", למחרוזת "rot", אולם לא למחרוזת "rut". כאשר רשימת תווים בביטוי רגולרי מוקפת באמצעות סוגריים מרובעים, כל תו ברשימה זו מתאים לתבנית.

לפניך דוגמה המראה כיצד תוכל להשתמש בביטוי רגולרי זה בדף ASP:

```
<%
myString = "The rat ate the rotting cheese."
Set myRegExp = New RegExp
myRegExp.Global = TRUE
myRegExp.Pattern = "r[ao]t"
Set myMatches = myRegExp.Execute( myString )
```

```
for each thing in myMatches
    Response.Write thing & "<BR>"
next
%>
```

תסריט זה יוצר מופע של האובייקט RegExp. נקבעים שני מאפיינים של האובייקט. ראשית, המאפיין Global משמש כדי לציין האם יש לסרוק את כל המחרוזת או האם יש להפסיק את הסריקה לאחר מציאת ההתאמה הראשונה. המאפיין Pattern משמש כדי לקשר את הביטוי הרגולרי לאובייקט RegExp.

לאחר שהאובייקט RegExp אותחל, נקראת השיטה Execute כדי להחזיר התאמות כלשהן. ההתאמות מוקצות לאובייקט myMatches (מופע של האובייקט Match). הלולאה FOR...EACH משמשת למעבר סידרתי על אוסף ההתאמות המוכלות באובייקט myMatches ולהתאמת תבניות ההתאמה בחלון הדפדפן. במקרה זה, מוצגות שתי המחרוזות "rat" ו-"rot".

ביטויים רגולריים הם בעלי עוצמה רבה. יישומם של ביטויים רגולריים ב-VBScript 5.0 כולל מאפיינים מיוחדים להתאמה לגבולות מילה, להתאמה לתווי רווח-לבן (Whitespace) ולהתאמה לרצף של תווים חוזרים.

## הסתרת קוד עם Microsoft Script Encoder

לאחרונה, שיחררה Microsoft את הגירסה הראשונה של Microsoft Script Encoder (http://msdn.microsoft.com/scripting). תוכנית שירות זו יכולה לשמש להגנת קוד המקור של תסריטי ASP.

**טיפ:**

החברה ServerObjects ייצרה כלי דומה לתוכנית השירות Microsoft Script Encoder הנקרא AspCodeLock. הוא נעזר ברכיב כדי לקודד ולהצפין דפי ASP בזיכרון המחשב. למידע נוסף, בקר באתר http://www.serverobjects.com.

אחד היתרונות של תסריטי ASP הוא שניתן ליצור תסריט ASP מבלי להדר אותו. החיסרון הוא שאם תפיץ את התסריטים שלך, כל אחד יוכל לראות את קוד המקור שלהם. Microsoft Script Encoder פותר בעיה זו באמצעות תרגום של תסריטי ASP לתבנית אשר אינה יכולה להיקרא על ידי עין אנושית.

לפניך דוגמה של דף ASP לפני הצפנה:

```
<HTML>
<HEAD><TITLE>Secret</TITLE></HEAD>
<BODY>
<%
' All this information is confidential
Set Con = Server.CreateObject(" ADODB.Connetction" )
Con.Open "DSN=myDSN;UID=sa;PWD=popcorn"
%>
</BODY>
</HTML>
```

ועתה, הנה דוגמה של אותו התסריט לאחר שהוצפן:

```
<%@ LANGUAGE = VBScript.Encode %>
<HTML>
<HEAD><TITLE>Secret</TITLE></HEAD>
<BODY>
<%#@~^kAAAAA==@#@&B~zV^PY4kk~k     0GdslOrKxPbdP1Wx6r[+
YGTt#@~w@2[]Y~YZZR{PyTgg/+. []Mg4~22#R/W   F55/@#YdlL;       xn^DkW
 EP*@#@&;GUR}w[]x~ghg'\FGt1WX1(9{/4rRRfxwK121W.        j@#@&^#~@%>
</BODY>
</HTML>
```

כדי להשתמש ב-Microsoft Script Encoder, פתח חלון DOS והפעל משורת הפקודה את תוכנית השירות הנקראת screnc.exe. לדוגמה, כדי להצפין את הקובץ secret.asp לקובץ בשם supersecret.asp, השתמש בפקודה:

```
screnc secret.asp supersecret.asp
```

לאחר שדף ASP הוצפן, תוכל להשתמש בו באתר האינטרנט שלך באופן רגיל. כיון שדף ASP מופעל בשרת, הצפנת דף ASP אינה יוצרת בעיות תאימות לדפדפן. בכל מקרה, אם התסריט כולל תסריטי צד-לקוח, הוא יפעל רק עם Internet Explorer 5.0.

**אזהרה:**

Microsoft מזהירה ש- Microsoft Script Encoder מונע רק קריאה מקרית של קוד המקור. אדם נחוש בדעתו יכול לפענח את הקבצים המוצפנים. אל תשתמש ב- Microsoft Script Encoder להגנה על מידע חסוי מאוד.

# הרחבות לאובייקטים המובנים של ASP

הגירסה של ASP הנכללת עם IIS 5.0 כוללת מספר שינויים חשובים באובייקטים המובנים של ASP. הגירסה החדשה של ASP כוללת שיטות מורחבות לשליטה בביצוע התוכנית, טיפול טוב יותר בשגיאות, ובאובייקטי Session ו-Application מורחבים. התכונות החדשות לביצוע תוכנית חשובות במיוחד ויתוארו בפירוט בסעיפים הבאים.

## שליטה בביצוע תוכנית

נניח שעליך להציג אחד משני דפי ASP, תלוי במידע אשר נשלח על ידי משתמש בטופס HTML. לפני IIS 5.0, היו רק שתי שיטות לבצע זאת: ניתן היה להשתמש בשיטה Redirect של האובייקט Server כדי לכוון מחדש משתמש לדף ASP אחר, או שניתן היה להציג אחד משני דפי ASP הנכללים עם הפניית INCLUDE#. לכל אחת משיטות אלו חסרונות רציניים.

השיטה Redirect של האובייקט Response פועלת על ידי שליחת כותרת (Header) חוזרת לדפדפן כדי לומר לו לבקש דף נוסף. שיטה זו לאחזור דף נוסף היא שיטה איטית כיון שהדפדפן חייב לבצע שתי בקשות כדי לקבל דף אחד. חמור מכך, דפדפנים מסוימים יתעלמו מההוראה לאחזר את הדף החדש באופן אוטומטי ובמקום זאת יציגו את ההודעה "Object Has Moved".

גם לשיטה הכוללת מספר דפי ASP בדף ASP בודד באמצעות הפניית INCLUDE# בעיות משלה. כאשר כוללים בדף ASP מספר דפי ASP, IIS ישתול את כל הדפים המוכללים לפני ביצוע עיבוד תסריט כלשהו. המשמעות היא, שתקבל דפי ASP מנופחים אשר גוזלים זיכרון שרת יקר ערך.

למרבה המזל, Microsoft סיפקה פתרון אלגנטי מאוד עם IIS 5.0. למעשה היא סיפקה שני פתרונות טובים. באמצעות השימוש בשיטה החדשה Transfer של האובייקט Server, תוכל להפנות משתמש לדף חדש בשרת, במקום בדפדפן. שימוש בשיטה החדשה Execute של האובייקט Server, יאפשר לך להפעיל תסריט בדף ASP אחד מתוך דף ASP אחר.

## שיפור בהפניית דפים

לשיטה החדשה Transfer של האובייקט Server תוצאות זהות לחלוטין כמו לשיטה Redirect, אך ללא אף אחד מחסרונותיה של זו. השיטה Transfer מפנה משתמש לדף אינטרנט אחר, מבלי לאלץ את הדפדפן לבקש דף חדש. כל הפעולות מתבצעות בשרת האינטרנט.

לדוגמה, נניח שברצונך לאלץ משתמש להקליד את שם המשתמש והסיסמה שלו לפני קבלת גישה לדף מסוים. להלן דוגמה לטופס HTML המבקש נתונים אלה:

```
<HTML>
<HEAD><TITLE>Simple Form</TITLE></HEAD>
<BODY>
```

```
<FORM method="post" action="result.asp">
<BR>Enter your name:
<inupt name="username">
<BR>Enter your password:
<input name="password">
<input type="submit" value="Okay">
</FORM>

</BODY>
</HTML>
```

טופס HTML זה נשלח לדף ASP בשם result.asp. בתוך result.asp תוכל לבדוק האם שם המשתמש והסיסמה אכן נשלחו, ואם לא, תוכל להפנות את המשתמש באופן אוטומטי חזרה לדף המקורי. להלן דוגמה כיצד לבצע זאת:

```
<%
username = TRIM( Request( "username" ) )
password = TRIM( Request( "password" ) )
if username = " " or password = " " then
    Server.Transfer "form.asp"
end if
%>
<HTML>
<HEAD><TITLE>Thanks</TITLE></HEAD>
<BODY>

Thanks for entering your username and password!

</BODY>
</HTML>
```

בדוגמה זו, משמשת השיטה Transfer של האובייקט Server להפניית המשתמש חזרה לטופס ה-HTML המקורי, אם שם המשתמש והסיסמה לא נשלחו. כאשר נקראת השיטה Transfer, נעצר הביצוע של התסריט הנוכחי. יתר דף ה-ASP לא יוצג לעולם, ובמקומו יוצג הדף החדש.

יתרון משמעותי נוסף של השיטה Transfer על פני השיטה Redirect היא שתוכנו של האוסף Request אינו הולך לאיבוד בעת מעברו בין הדפים. לדוגמה, כאשר נשלחים הנתונים שבטופס לדף result.asp והמשתמש מועבר חזרה לדף form.asp, הנתונים המקוריים שנשלחו על ידי המשתמש עדיין זמינים בדף form.asp. זה אומר שניתן למלא שוב את כל נתוני הטופס באופן אוטומטי:

```
<%
username = TRIM( Request( "username" ) )
password = TRIM( Request( "password" ) )
%>
<HTML>
<HEAD><TITLE>Simple Form</TITLE></HEAD>
<BODY>
```

```
<FORM method="post" action="result.asp">
<BR>Enter your name:
<inupt name="username" value="<%=username%>">
<BR>Enter your password:
<input name="password" value="<%=password%>">
<input type="submit" value="Okay">
</FORM>

</BODY>
</HTML>
```

התסריט הקודם הוא גירסה משוכתבת של הדף form.asp המקורי. גם שם המשתמש וגם הסיסמה מעודכנים בערכים מהאוסף Request. אם המשתמש שולח את הטופס עם שם המשתמש אולם שכח למלא את הסיסמה, הוא יועבר בחזרה לדף זה, ושם המשתמש שמולא על ידו במקור יופיע שנית באופן אוטומטי.

**הערה:**

אולי תהיה סקרן לדעת מה יתרחש אם תשתמש בשיטה Transfer כדי להעביר דף ASP בחזרה לעצמו. האם הדף יילכד בלולאה אין-סופית? האם המחשב יתחיל להינתך (to Melt)? למעשה, התסריט יועבר בחזרה לעצמו 115 פעמים, ואז ייעצר. תוכל לבחון תוצאה מוזרה זו באמצעות דף ה-ASP הבא הנקרא myself.asp:

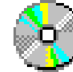
```
<%
response.write application( "count" ) & "<BR>"
application( "count" ) = application( "count" ) + 1
server.transfer "myself.asp"
%>
```

קיים מספר מוגבל של מצבים בהם יהיה זה הגיוני יותר להשתמש בשיטה Response.Redirect מאשר בשיטה Response.Transfer. לא ניתן להשתמש בשיטה Response.Transfer כדי להפנות משתמש לדף באתר אינטרנט שונה. השיטה Transfer יכולה להפנות משתמשים אך ורק לדף אחר באותו אתר.

בנוסף, לא ניתן להשתמש בשיטה Response.Transfer כדי להעביר משתמש ליישום ASP שונה. תוכל להעביר משתמש לדף הקיים בתוך יישום ASP אחר, אולם לא תוכל לזכות בגישה למשתני היישום של היישום השני.

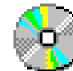
באופן כללי, עליך להימנע משימוש בשיטה Redirect לטובת השימוש בשיטה Transfer. השיטה Transfer מהירה יותר, וכיון שהיא מתבצעת רק בשרת, היא תואמת לכל סוגי הדפדפנים.

# ביצוע דפי ASP



IIS 5.0 כולל שיטה שנייה להעברת ביצוע תסריט מדף ASP אחד לשני. תוכל להשתמש בשיטה Execute של האובייקט Server כדי לבצע תסריט בדף ASP אחד מתוך דף ASP אחר. התייחס לדף ASP הבא, הנקרא Active1.asp (נמצא בתקליטור תחת השם Active1.asp):

```
<!-- Active1.asp -->
<HTML>
<HEAD><TITLE>Messages</ TITLE ></ HEAD >
<BODY>

Executing another script:
<% Server.Execute "active2.asp" %>
<BR>All Done!

</BODY >
</HTML>
```



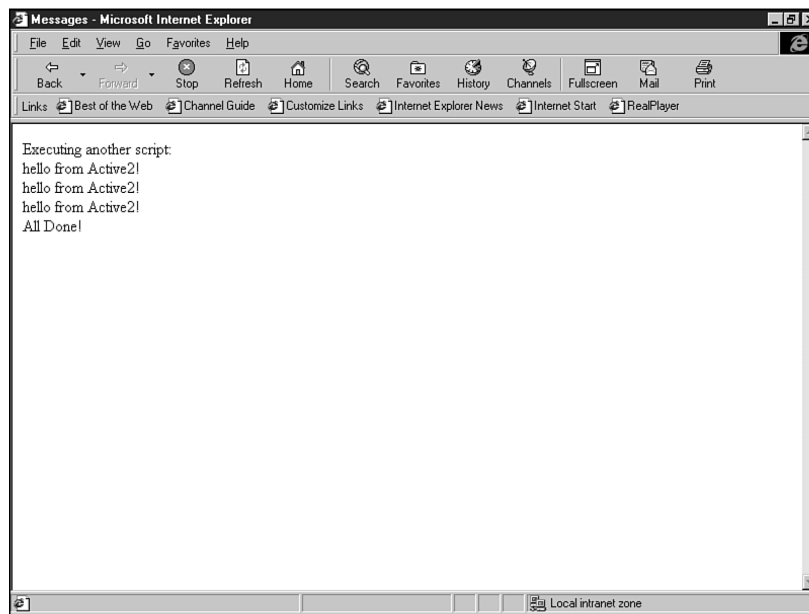
דף ASP זה מבצע דף ASP שני הנקרא Active2.asp (נמצא בתקליטור תחת השם Active2.asp). כאשר מבוצע התסריט Active2.asp, מוצגת הודעה בחלון הדפדפן שלוש פעמים. לאחר גמר ביצוע התסריט Active2.asp, מוצגת יתרת התסריט Active1.asp (ראה להלן תרשים 26.4).

```
<!--  Active2.asp -->
<%
for i = 1 to 3
   Response.Write "<br>hello from Active2!"
next
%>
```

אולי תופתע ללמוד שמשתני תסריט אינם מועברים בין שני התסריטים. אם מוצב ערך למשתנה בתסריט Active1.asp, אזי אי אפשר לאחזר את ערכו של המשתנה מתוך התסריט Active2.asp. אם עליך להעביר משתנה בין שני תסריטים, אתה חייב להשתמש במשתנה Application או במשתנה Session, או להעביר את הערך בתוך מחרוזת שאילתה.

לדוגמה, התסריט הבא מעביר הודעה מהתסריט outerscript.asp לתסריט innerscript.asp :

```
<! - - outerscript.asp - ->
<HTML>
<HEAD><TITLE>Massages</TITLE></HEAD>
<BODY>

Executing another script:
<%
Allplication( "message" ) = "hello!"
Server.Execute "innerscript.asp" %>
<BR>All Done!

</BODY>
</HTML>
```




**תרשים 26.4** פלט התסריט Active1.asp.

ההודעה "hello!" מועברת בתוך משתנה Application כאשר נקרא התסריט innerscript.asp. כאשר התסריט מבוצע, ערכו של המשתנה Application מוצג בצורה נכונה.

```
<! - - innerscript.asp - ->
<%=Application( "message" )%>
```

**הערה:**

תוכל להיות רשלני יותר בשימוש במשתני Application עם IIS 5.0 לעומת השימוש בהם בגרסאות הקודמות של IIS. שלא כמו בגרסאות קודמות של IIS, IIS 5.0 כולל שתי שיטות להסרת משתני Application מהזיכרון. באמצעות שימוש בשיטה Remove של האובייקט Application, ניתן להסיר משתנה Application בודד, ובאמצעות שימוש בשיטה RemoveAll תוכל להסיר את כולם. שיטות דומות קיימות עבור גם האובייקט Session.

# הרחבות לרכיבים המותקנים של ASP

Microsoft טוענת שביצועיהם של רכיבי ה-ASP הנכללים עם IIS 5.0 שופרו באופן ניכר, כך שגודלם ניתן לשינוי בקלות רבה יותר. מלבד שדרוגי ביצועים אלה, נעשו ברכיבים רק שינויים קטנים ביותר. Microsoft הוסיפה רכיב חדש יחיד לגירסה החדשה של IIS: הרכיב Logging Utility.

## הרכיב Logging Utility

הרכיב Logging Utility מאפשר גישה לקבצי יומן השרת מתוך תסריטי ASP. אם עליך לחולל נתונים סטטיסטיים מותאמים אישית על אופן השימוש באתר שלך, רכיב זה יכול להיות רב ערך.

לדוגמה, אם אתה מציג כרזות פרסום באתר שלך, תוכל להשתמש ברכיב זה כדי לאחזר ולהציג את מספר הפעמים בהם נראתה כרזת הפרסום ביום מסוים. אם עליך לשפר את ביצועי האתר, הרכיב Logging Utility יכול לאחזר נתונים סטטיסטיים לגבי דפי ה-ASP הפועלים בצורה האיטית ביותר. לבסוף, הרכיב יכול לשמש לאיסוף מידע על הדפים הנצפים ביותר באתר.

ישנן שתי הגבלות לשימוש ברכיב Logging Utility. ראשית, הרכיב פועל רק על ארבעה פורמטים של קבצי יומן הנכללים עם ASP. הוא לא יפעל עם רישום יומן ODBC (בכניסה למסד נתונים), והוא לא יפעל עם פורמטים מותאמים אישית של קבצי יומן המפותחים על ידי חברות של צד שלישי.

בנוסף, כדי להשתמש ב-Logging Utility עליך להיות מאומת בשרת האינטרנט שלך. במילים אחרות, אינך יכול לגשת לדף ASP המשתמש ברכיב זה באמצעות חשבון משתמש אנונימי. כדי למנוע גישה אנונימית לקובץ, הפעל את Internet Services Manager, פתח את דף המאפיינים עבור הקובץ, בחר בכרטיסיה File Security, ונטרל את האפשרות Allow Anonymous Access.

# סיכום

פרק זה סיפק סקירה קצרה על כמה מהתכונות החשובות החדשות של Windows 2000, IIS 5.0, VBScript 5.0 ו-ASP. בחלקו הראשון של הפרק, למדת על התכונות החדשות של IIS 5.0 המספקות דחיסת HTTP, אפשרויות אבטחה חדשות, דפי ASP ללא תסריטים, והפרוטוקול DAV (Distributed Authoring and Versioning). בחלק השני, למדת על התכונות וההרחבות החדשות ל- VBScript 5.0. בחלק השלישי, למדת על התכונות החדשות לבקרת זרימת תוכנית השייכות לאובייקט Server של ASP. לבסוף, בחלק האחרון, הובאה לפניך סקירה קצרה של הרכיב החדש הנכלל עם ASP: הרכיב Logging Utility.

# חלק 7

# דוגמאות ליישומי ASP

**בחלק זה:**

- **יצירת פורום דיון**
- **יצירת חנות מקוונת**
- **יצירת אתר לחיפוש עבודה**

# פרק 27

# יצירת פורום דיון

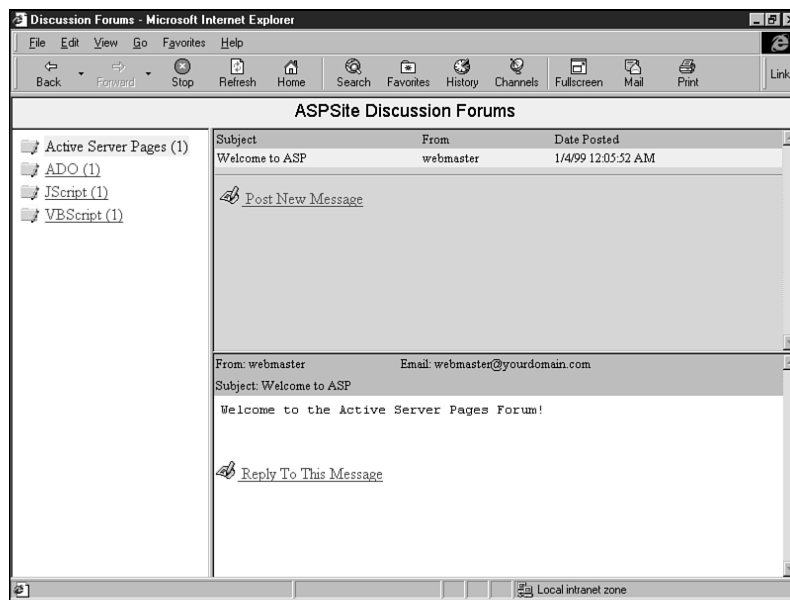
**בפרק זה:**

- **סקירת הפרויקט**
- **בניית פורום הדיון**
- **הצגת גוף ההודעה**
- **הרחבת פורום הדיון**

למשך תקופה ארוכה אירח האתר המלווה ספר זה פורום דיונים המוקדש רובו ככולו לפיתוח ותכנות דפי ASP. פורום זה זכה להצלחה גדולה ולכמות גדולה של משתתפים. אחת השאלות הפופולריות ביותר שהוצגו בפורום זה, וזה לא מפתיע כלל, היתה לגבי האופן בו נכתב פורום הדיון עצמו. בפרק זה יתגלו בפניך כל הסודות הכמוסים של פורום הדיון שבאתר ASPSite. בדפים הבאים תלמד כיצד ליצור פורום דיון משל עצמך, אותו תוכל לשלב באתר האינטרנט שלך.

פורום הדיון המתואר בפרק זה כולל מיגוון רחב של מאפיינים מתקדמים:

- הוא תומך בריבוי אזורי דיון. למשל, הפורום שב-ASPSite כולל אזורים נפרדים המוקדשים לתכנות באמצעות ADO, SQL, JScript ו-VBScript.
- הוא תומך בקישור נושאי דיון (Threading). אם מישהו מפרסם שאלה, יכולים מספר משתמשים להשיב על אותה שאלה וכל תשובה תופיע ישירות מתחת לשאלה המקורית.
- הוא עושה שימוש נרחב במסגרות (Frames). רשימת אזורי הדיון, רשימת ההודעות וההודעה הנוכחית, כולם מופיעים כל אחד במסגרת משל עצמו (ראה תרשים 27.1). דבר זה מקל על הניווט בין ההודעות ואזורי הדיון.
- פורום הדיון משתמש רק בתסריטי ASP של צד-שרת. מאחר והוא אינו מסתמך על אף תכונה של הדפדפן, פורום דיון כגון זה תואם לכל סוגי הדפדפנים התומכים בהצגת מסגרות.




**תרשים 27.1** פורום הדיון

# סקירת הפרויקט

כמו כל פרויקט תכנות, כך גם פורום הדיון מציב בפני המפתח מספר אתגרים. יש להתגבר על שתי בעיות עיקריות: הבעיה בסנכרון תוכן המסגרות השונות והבעיה ליצור דיונים מקושרים (Threaded Discussions).

## סנכרון מסגרות

פורום הדיון כולל שלוש מסגרות HTML אשר חייבות להיות מסונכרנות ביניהן. המסגרת השמאלית, נקראת forumlist, מכילה את רשימת אזורי הדיון הקיימים. כאשר מישהו לוחץ על אזור דיון מסוים צריכות שתי המסגרות האחרות להציג את ההודעות של הפורום הנבחר. המסגרת העליונה, נקראת messagelist, מכילה את רשימת ההודעות. כאשר המשתמש לוחץ על הודעה מציגה המסגרת התחתונה, הנקראת message, את תוכן ההודעה (Message Body).

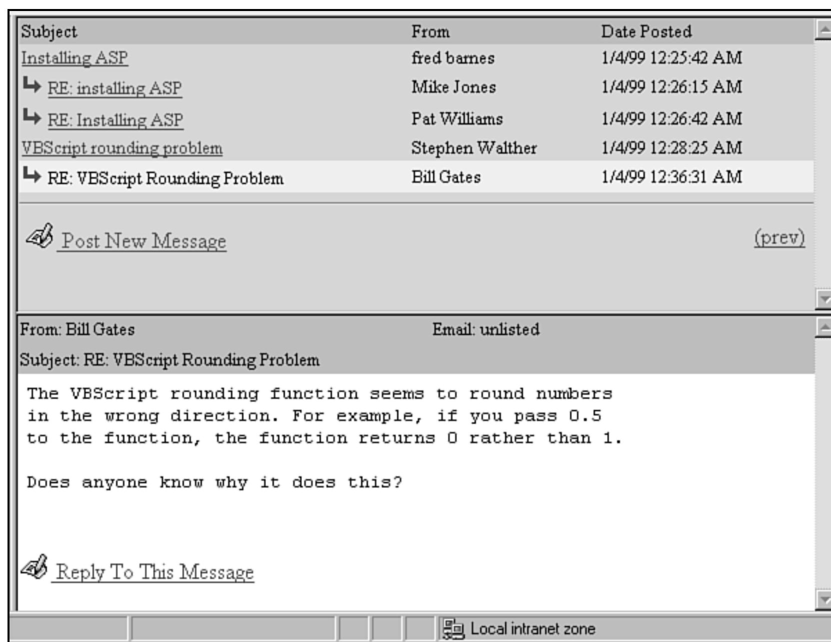
סנכרון תוכנן של המסגרות בפורום דיון מציבה בפני מפתח הפורום בעיה. בדרך כלל, כאשר אתה לוחץ על קישור היפר-טקסט, נטען דף אחד בלבד. אבל, בפורום הדיון, כאשר אתה לוחץ על שם של פורום ברשימת הפורומים הקיימים או על הודעה כלשהי ברשימת ההודעות, צריך להתעדכן תוכנה של יותר מאשר מסגרת יחידה.

לדוגמה, ההודעות בפורום הדיון מוצגות בשתי מסגרות. המסגרת העליונה מציגה את רשימת ההודעות, ואילו המסגרת התחתונה מציגה את תוכן ההודעה הנבחרת כרגע. כאשר המשתמש בוחר לצפות בהודעה חדשה על ידי לחיצה על הקישור אליה במסגרת העליונה, משתנה המסגרת העליונה כך שההודעה הנוכחית מוארת (צבע הרקע של הכותרת שלה משתנה), והמסגרת התחתונה משתנה כך שתציג את תוכן ההודעה שנבחרה (ראה תרשים 27.2).

פתרון אחד לבעיה כגון זו של סנכרון מספר מסגרות בו-זמנית הוא להשתמש בשפת תסריט של צד-לקוח, כגון שפת JavaScript. לדוגמה, הצהרת JavaScript הבאה יכולה לשמש כדי לשנות את תוכנה של מסגרת אחת מתוך מסגרת אחרת:

```
<SCRIPT LANGUAGE="JavaScript">
   parent.message.location.href="message.asp?m=2";
</SCRIPT>
```

הצהרה זו טוענת את הדף message.asp למסגרת בשם message. אם תסריט זה מוצב בחלקו העליון של פורום הדיון, אמורה המסגרת התחתונה להסתנכרן באופן אוטומטי עם ההודעה הנבחרת הנוכחית. בהצהרה זו מועבר מזהה ההודעה (Message ID) שאמורה להיות מוצגת בתור משתנה URL (URL Variable).




**תרשים 27.2** סנכרון מסגרות

אולם, לשימוש בתסריט של צד-לקוח יש חיסרון אחד משמעותי. לא כל הדפדפנים תומכים תמיכה מלאה ב-JavaScript, וכשהדפדפן אינו תומך ב-JavaScript השגיאה עלולה להיות מכוערת למדי. תסריטי ASP, לעומת זאת, אינם סומכים על מאפיין כלשהו של הדפדפן.

פורום הדיון אינו נעזר בתסריטי JavaScript לצורך סנכרון המסגרות. במקום זאת, המסגרות מתואמות על ידי טעינה מחדש של קובץ המסגרות (Frameset File) המכיל את המסגרות. הנה גירסה פשוטה יותר של קובץ המסגרות המכיל את המסגרות messagelist ו-message.

```
<!-- Messages Frameset File -->
<% messageID = TRIM( Request( "m" ) ) %>
<HTML>
<HEAD><TITLE> Messages </TITLE></HEAD>
<FRAMESET ROWS="*,*">
   <FRAME NAME="messagelist" SRC="messagelist.asp?m=<%=messageID%>" >
   <FRAME NAME="message" SRC=" message.asp?m=<%=messageID%>" >
</ FRAMESET>
</HTML>
```

כשהמשתמש לוחץ על קישור במסגרת messagelist, הקישור טוען מחדש את קובץ המסגרות הראשי. להלן דוגמה לקישור שעשוי להיראות במסגרת messagelist:

```
<A HREF="message.asp?f=ADO&m=21" TARGET="messages"
Setting Cursor Types?</A>
```

כאשר קובץ מסגרות ההודעות (Frameset file) נטען מחדש באמצעות קישור זה, נטענות מחדש גם המסגרות messagelist ו-message. מזהה ההודעה הנוכחית מועבר לשתי המסגרות, באמצעות המשתנה m של URL. במילים אחרות, מספר מסגרות מעודכנות פעם אחת באמצעות קישור יחיד.

שיטה זו של עדכון וסנכרון תוכן של מספר מסגרות תואמת לכל הדפדפנים התומכים במסגרות. גישה זו משמשת את פורום הדיון הן לתיאום בין רשימת ההודעות ובין ההודעה הנבחרת הנוכחית, והן לתיאום בין רשימת הפורומים לבין רשימת ההודעות של הפורום הנבחר.

**הערה:**

תוכל להשתמש בשיטה המתוארת כאן לשינוי קובץ מסגרות (frameset file), כדי לדמות קבצי INCLUDE דינמיים. אינך יכול לשייך משתנה להפניית #INCLUDE של ASP, אך אתה כן יכול לשייך משתנה לתכונת SRC של מסגרת, מתוך תסריט ASP. להלן תסריט לדוגמה המציג דפים שונים באופן אקראי בתוך שתי מסגרות (תמצא את הקובץ dynamicframe.asp בתקליטור המצורף לספר זה).

```
<%
randomize
if rnd()  > 0.5 then
   framesrc = "http://www.microsoft.com"
else
   framesrc = "http://www.netscape.com"
end if
%>
<HTML>
<FRAMESET ROWS="*,*">
<FRAME SRC="<%=framesrc%>">
<FRAME SRC="<%=framesrc%>">
</FRAMESET>
</HTML>
```

# יצירת דיונים מקושרים

האתגר השני בבניית פורום דיון היה למצוא דרך יעילה לאחסון והצגת דיון מקושר (Threaded Discussion). הבעיה בדיון מקושר היא שתגובות חדשות יכולות להיות מוצבות באמצע רשימת ההודעות בכל זמן. למשל, מישהו פרסם הודעה בתאריך 29 ביולי 2000, ומישהו אחר כתב תגובה לאותה הודעה שנה מאוחר יותר. העובדה שהודעות יכולות לקבל תגובה יוצרת מצב בו ההודעות אינן יכולות להיות מאוחסנות ומוצגות בסדר בו הן פורסמו.

כל ההודעות מאוחסנות בטבלה יחידה הנקראת messages. טבלה זו כוללת עמודות עבור כותב ההודעה (Auther), נושא ההודעה (Subject), יום הוספת/כתיבת ההודעה (Entry Date) ותוכן ההודעה (Body). הטבלה גם כוללת עמודה המייצגת את מספר

הסודר של ההודעה. מספר הסודר (Order Number) של ההודעה קובע את הסדר על פיו תוצגנה ההודעות.

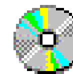
גישה אחת לטיפול בתגובות חדשות להודעות היא פשוט לעדכן את מספר הסודר של ההודעות, בכל פעם שנכנסת תגובה חדשה. למשל, אם פורסמה תגובה להודעה מספר 32, מספר הסודר של כל ההודעות שמספרן גדול מ- 32 יקודם באחד, והתגובה החדשה תאוחסן עם מספר סודר 33. השימוש בשיטה זו ייצור התאמה של אחד לאחד בין מספרי הסודר של ההודעות והסדר בו הן מוצגות.

הבעיה היא שגישה זו יוצרת עומס על בסיס הנתונים. אם מפורסמת תגובה להודעה הממוקמת באמצע רשימת ההודעות, הדבר אומר כי מספרן של מחצית מההודעות שברשימה צריך להיות מעודכן למספר סודר חדש.

הפורום שלנו נוקט בגישה שונה. בכל פעם שמפורסמת תגובה חדשה להודעה קיימת, מאוחסנת התגובה באותו מספר סודר של ההודעה המקורית, זו אשר לגביה פורסמה התגובה. למשל, כל התגובות להודעה שמספרה 32 מקבלות גם הן את מספר הסודר 32. רשומת התאריך של ההודעה היא זו המשמשת כעת להצגת התגובות בסדר הנכון.

יתרונה של שיטה זו לאחסון ההודעות טמון בכך שהיא אינה גורמת לעומס על בסיס הנתונים. כאשר מפורסמת תגובה חדשה אין צורך לעדכן איזו מההודעות הקיימות. כל תגובה משויכת להודעה המקורית המתאימה וסדר ההודעות הנכון נשמר.

# בניית פורום הדיון



כדי להשתמש בפורום הדיון תצטרך להעתיק את כל דפי ASP שלו מהתיקיה המתאימה שבתקליטור המצורף לספר זה, לתיקיה בשרת שלך. חובה ששרת האינטרנט שלך יוכל לגשת לתיקיה זו.

כעת לאחר שהעתקת את כל הקבצים, עליך לערוך שני שינויים בקובץ ASP בשם forumfuncs.asp. ראשית, עליך לשנות את הפניית #INCLUDE שבראש הדף כך שיצביע על הקובץ ADOVBS.INC שבשרת שלך (הקובץ ADOVBS.INC הותקן באופן אוטומטי בשרת שלך בעת התקנת IIS).

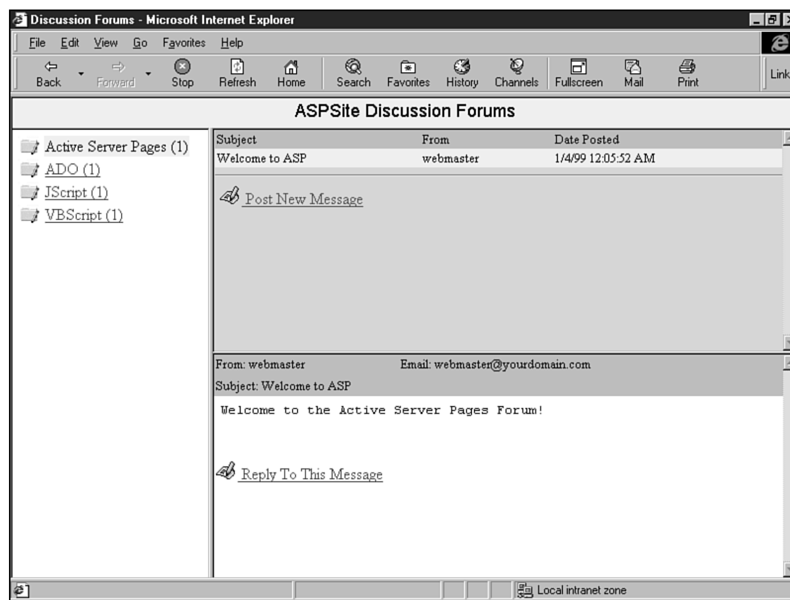
בנוסף, תצטרך לשנות את מחרוזת החיבור לבסיס הנתונים (Database Connection String) המשמשת עם פורום הדיון. מחרוזת החיבור לבסיס הנתונים נקבעת על ידי קבוע (Constant) בשם dbCon המופיע בחלקו העליון של הקובץ forumfuncs.asp. עליך לספק שם קובץ Data Link חוקי עבור בסיס הנתונים בו תשתמש כדי לאחסן את הודעות פורום הדיון.

**הערה:**

למידע נוסף אודות מחרוזת החיבור לבסיס הנתונים (Database Connection String), פנה לפרק 12.

לסיום, עליך להתקין את טבלת בסיס הנתונים והשגרות המאוחסנות המשמשות את הפורום. כדי לעשות זאת, טען את תסריט SQL בשם installforums.sql מהתקליטור המצורף לספר אל ISQL /w Microsoft או Microsoft Query Analyser, בחר בסיס נתונים והפעל את התסריט. כשתפעיל את התסריט, תיווצר באופן אוטומטי ערכת פורומים.

כדי לצפות בפורום דיון, טען את הקובץ forums.asp לדפדפן האינטרנט שלך. אם הכל הותקן כראוי, אמור פורום הדיון להופיע כפי שהוא נראה בתרשים 27.3.




**תרשים 27.3** פורום הדיון המותקן

## היחסים בין קבצי הפורום

פורום הדיון כולל 10 קבצי ASP ייחודיים, כולם זמינים בתקליטור המצורף לספר זה. היחסים בין הקבצים הללו הם מורכבים, כך שסעיף זה מציג סקירה קצרה לגבי מהות היחסים.

- forum.asp - הקובץ אותו אתה טוען לדפדפן שלך כדי לצפות בפורום הדיון. forum.asp הוא קובץ מסגרות (frameset) המגדיר שתי מסגרות. המסגרת העליונה מכילה את topbar.asp. המסגרת התחתונה מכילה את קובץ המסגרות content.asp.
- topbar.asp - קובץ זה מציג את הכרזה המופיעה בחלקו העליון של המסך. הלוגו ASPSite Discussion Forums מוצג כאן. תוכל לשנות לוגו זה לכל לוגו אחר שתבחר.
- content.asp - קובץ מסגרות (frameset) זה מכיל את המסגרת forumlist.asp ואת קובץ המסגרות message.asp. כאשר מפורסמת הודעה חדשה, נטען מחדש קובץ מסגרות זה, כדי שכל מסגרות הצאצאים שלו ייטענו מחדש גם הן.
- forumlist.asp - קובץ זה משמש להצגת כל אזורי הדיונים. הוא מפורט ביתר הרחבה בסעיף **הצגת רשימת הפורומים** שבהמשך הפרק.
- messages.asp - קובץ מסגרות המגדיר את המסגרות messagelist.asp ו-message.asp. כאשר נבחרת הודעה חדשה קובץ זה נטען מחדש כדי שכל מסגרות הצאצאים שלו ייטענו מחדש גם הן כאשר ההודעה הנכונה נבחרת.
- messagelist.asp - מציג את רשימת ההודעות שבאזור הדיון הנבחר. קובץ זה מתואר ביתר הרחבה בסעיף **הצגת רשימת ההודעות** שבהמשך פרק זה.
- message.asp - משמש להצגת תוכן ההודעה עצמו. קובץ זה מתואר ביתר הרחבה בסעיף **הצגת גוף ההודעה** שבהמשך פרק זה.
- post.asp - מכיל את הטופס לשליחת הודעה חדשה. קובץ זה מתואר ביתר הרחבה בסעיף **הוספת הודעה חדשה** שבהמשך פרק זה.
- reply.asp - מכיל את הטופס המיועד להוספת תגובה להודעה קיימת. קובץ זה מתואר ביתר הרחבה בסעיף **תגובה להודעה** שבהמשך פרק זה.
- forumfuncs.asp - מכיל את כל הפונקציות והשגרות השכיחות בהן משתמשים הדפים האחרים.

## טבלת ההודעות

פורום הדיון משתמש בטבלת בסיס נתונים יחידה, הנקראת messages, כדי לאחסן בו את כל ההודעות המפורסמות בכל אזורי הדיון בפורום. טבלה זו נקראת טבלת ההודעות (Messages Table) והיא נוצרת באמצעות הצהרת CREATE TABLE של SQL, כפי שמוצג להלן בדוגמה הבאה:

```
CREATE TABLE dbo.messages (
    m_id int IDENTITY (1, 1) NOT NULL ,
    m_forumName varchar (30) NOT NULL ,
    m_subject varchar (30) NOT NULL ,
    m_username varchar (30) NOT NULL ,
    m_email varchar (70) NOT NULL ,
    m_entrydate datetime NOT NULL DEFAULT getDATE(),
    m_message text NULL ,
    m_ordernum int NULL ,
    m_reply bit NOT NULL
)
```

העמודה m_id משמשת לשיוך מזהה הודעה ייחודי (Unique Message ID) לכל הודעה (זו עמודת זהות). העמודה m_forumName משמשת לציון הפורום בו אמורה ההודעה להופיע, העמודה m_ordernum קובעת את סדר הצגת ההודעות והעמודה m_reply מציינת אם הודעה זו היא תגובה להודעה אחרת, או לא.

העמודה m_entrydate מכילה את התאריך והשעה בה נוספה ההודעה לטבלת ההודעות. שים לב שלעמודה יש ערך ברירת מחדל, getDATE(). הפונקציה getDATE() של SQL משייכת לעמודה את התאריך והשעה הנוכחיים באופן אוטומטי.

התוכן הממשי של ההודעה מאוחסן בעמודות m_subject, m_username, m_email ו-m_message, שהן נושא ההודעה, שם כותב ההודעה, כתובת הדואר האלקטרוני של כותב ההודעה וגוף ההודעה, בהתאמה.

## הצגת רשימת הפורומים

המסגרת השמאלית של פורום הדיון מציגה את כל אזורי הדיון הזמינים. כאשר אתה לוחץ על קישור כלשהו במסגרת זו מוצגות ההודעות המתאימות לאזור הדיון שנבחר. דף ASP בשם forumlist.asp משמש להצגת רשימת הפורומים.

רשימת הפורומים מאוחזרת תוך שימוש בשגרת SQL המאוחסנת הבאה:

```
create procedure getForums
as
select m_forumname, count( m_id ) theCount
from messages
group by m_forumname
order by m_forumname
```

שגרה מובנית זו מאחזרת את שמו של כל פורום מטבלת ההודעות. בנוסף מאוחזר גם מונה מספר ההודעות בכל פורום.

אם אתה מעוניין להוסיף פורום נוסף עליך פשוט להוסיף הודעה לטבלת ההודעות ובה שם הפורום החדש. למשל, הצהרת SQL הבאה מוסיפה פורום חדש בשם Other Topics :

```
insert messages (
    m_forumName,
    m_username,
    m_email,
    m_subject,
    m_message,
    m_ordernum,
    m_reply
    ) values (
    'Other Topics',
    'webmaster',
    'webmaster@yourdomain.com',
    'Welcome to Other Topics',
    'Welcome to Other Topics!',
    0,
    0
    )
```

כדי להסיר פורום, מחק בטבלת ההודעות את כל ההודעות המכילות את שם הפורום המיועד למחיקה. כדי לשנות שם של פורום, עדכן את תוכן העמודה forumName בשם הפורום החדש.

כאשר פורום הדיון נטען בפעם הראשונה נבחר באופן אוטומטי אזור הדיון המוגדר כברירת מחדל. פורום ברירת המחדל נקבע על ידי קבוע (Constant) בשם defaultForum המופיע בחלקו העליון של הקובץ content.asp. אם תשנה את שמות אזורי הדיון עליך לשנות גם את קבוע זה.

# הצגת רשימת ההודעות

המסגרת העליונה של פורום הדיון מציגה את רשימת ההודעות התואמות לפורום הנוכחי. ההודעות מופיעות על פי נושא ההודעה, כותב ההודעה והתאריך בו נוספה הודעה זו לפורום. בכל רגע נתון מוצגות עד 15 הודעות.

כאשר הפורום נטען לראשונה בדפדפן, מאוחזר באופן אוטומטי מזהה ההודעה (Message ID) של ההודעה האחרונה שפורסמה בפורום ברירת המחדל. דבר זה מתבצע על ידי שגרת SQL המאוחסנת הבאה :

```
CREATE PROCEDURE getLastMessage
(
    @forumName varchar( 30 ),
    @lastMessage int OUTPUT
)
```

```
AS
declare @maxOrderNum int, @maxDate datetime

select @maxOrderNum = max( m_ordernum )
from messages
where m_forumName = @forumName

select @maxDate = max( m_entrydate )
from messages
where m_ordernum = @maxOrderNum
and m_forumName = @forumName

select @lastmessage = m_id from messages
where m_ordernum = @maxOrdernum
and m_entrydate = @maxDate
and m_forumName = @forumName
```

שגרה זו מורכבת מכפי שאתה עשוי לצפות ממנה, מפני שיכול להיות שמזהה ההודעה של ההודעה האחרונה עשוי שלא להיות מזהה ההודעה הגבוה ביותר. אם פורסמה תגובה להודעה מוקדמת בפורום יהיה מזהה ההודעה של ההודעה האחרונה נמוך מזה של הודעת התגובה. כדי לעקוף בעיה כגון זו נעשה שימוש בפונקציה המצטברת MAX() של SQL, כדי לאחזר את מזהה ההודעה עם תאריך הכניסה הקרוב ביותר ומספר הסודר הגבוה ביותר בפורום הנוכחי.

לאחר שאוחזר מזהה הודעה, משמש קובץ ASP בשם messagelist.asp כדי להציג את רשימת ההודעות. רשימת ההודעות מאוחזרת מטבלת ההודעות, תוך שימוש בשגרת SQL המאוחסנת הבאה:

```
create procedure getMessages
(
    @forumName varchar( 30 ),
    @messageID int
)
as
declare @messageOrder int

select @messageOrder = m_ordernum
from messages
where m_id = @messageID

select
    m_id,
    m_ordernum,
    m_username,
    m_subject,
    m_entrydate,
    m_reply
    from messages
    where m_forumName = @forumName
```

```
    and m_ordernum > ( @messageOrder - 6 )
    and m_ordernum < ( @messageOrder + 16 )
    order by m_ordernum, m_entrydate
```

לשגרה מאוחסנת זו מועברים שני פרמטרים. שם הפורום הנוכחי ומזהה ההודעה הנוכחי. השגרה מאחזרת את מספר הסודר של ההודעה ממזהה ההודעה ובוחרת את כל ההודעות להן יש מספר סודר קרוב למספר סודר (Order Number) זה.

יותר מהודעה אחת בטבלת ההודעות יכולה להיות בעלת אותו מספר סודר. למשל, לכל התגובות להודעה מסוימת יהיה אותו מספר סודר כמו זה של ההודעה המקורית. דבר זה אומר שהשגרה getMessages יכולה להשיב כל מספר של הודעות.

שתי השגרות, getMessages ו-showMessages, אותן תמצא בקובץ forumfuncs.asp משמשות להצגה בפועל של רשימת ההודעות. השגרה getMessages מציבה את כל ההודעות שאוחזרו על ידי השגרה המאוחסנת של getMessages SQL לתוך מערך (Array). הדבר נעשה באמצעות השיטה GetRows() של האובייקט RecordSet:

```
theArray = RS.getRows()
```

רשימת ההודעות מועברת מ-Recordset למערך כך שרק 15 הודעות יוצגו בכל רגע נתון. השגרה getMessages מוצאת את מספר האינדקס של ההודעה הנבחרת הנוכחית על ידי ביצוע איטרציה על תוכן המערך. לאחר שההודעה הנבחרת הנוכחית מאותרת, נקבעת ההודעה הראשונה שיש להציג על ידי חיסור 4 ממיקום האינדקס של ההודעה הנבחרת הנוכחית. התסריט הבא מאתר את האינדקס של ההודעה הראשונה אותה יש להציג:

```
for k = 0 to UBOUND( theArray, 2)
    if cLNG( theArray( m_id, k ) ) = cLNG( theMessageID ) then
        theFirstMessage = k - 4
        if theFirstMessage < 0 then theFirstMessage = 0
        exit for
    end if
next
```

לאחר שנמצא אינדקס ההודעה הראשונה מוצגת רשימת 15 ההודעות באמצעות השגרה showMessages. השגרה showMessages מציגה את ההודעות בטבלה. כל ההודעות, למעט ההודעה הנבחרת הנוכחית, מוצגות באמצעות צבע הרקע המוגדר כברירת מחדל.

כל הודעה ברשימה, למעט ההודעה הנבחרת, מוצגת כקישור (Link). הקישור כולל משתנה URL עבור שם הפורום ומשתנה URL עבור מזהה ההודעה. להלן דוגמה:

```
<A HREF="message.asp?f=ADO&m_21" TARGET="messages">
Setting Cursor Types? </A>
```

כאשר אתה לוחץ על קישור נטען קובץ המסגרות מחדש, וכל תהליך הצגת רשימת ההודעות מתחיל מראשיתו.

# הצגת גוף ההודעה

המסגרת התחתונה של פורום הדיון מציגה את גוף ההודעה הנבחרת הנוכחית. המסגרת התחתונה מציגה דף ASP בשם message.asp. לאחזור הודעה מטבלת ההודעות נעשה שימוש בתסריט הבא:

```
''''''''''''''''''''''''''''''''''''''''''''''''''''
'  Retrieve The Message
''''''''''''''''''''''''''''''''''''''''''''''''''''
MySQL = "select * from messages where m_id=" & messageID
Set RS = Server.CreateObject( "ADODB.RecordSet" )
RS.Open MySQL, Con, adOpenStatic
```

**טיפ:**

שים לב שסמן ברירת המחדל forward-only אינו משמש לאחזור ההודעה. במקומו נפתח סמן סטטי. כאשר מאוחזרות עמודות טקסט אל לך להשתמש בסמן forward-only, מפני שלפעמים עלול תוכן עמודת הטקסט להיקטע (קרא על סוגי הסמנים בפרק 13).

גוף (תוכן) ההודעה מוצג באמצעות התסריט הבא:

```
<TABLE WIDTH="100%" CELLPADDING=4 BORDER=0>
<TR>
   <TD>
   <PRE><%=Server.HTMLEncode( RS( "m_message" ) )%></PRE>
   </TD>
</TR>
</TABLE>
```

התגית <PRE> משמשת לשמירת תווי החזרה (Carriage Return, המקביל להקשה על Enter) ותווי רווח בהודעה המקורית. ללא תגית זו תופיע ההודעה ללא הריווח המקורי שלה, כשורה ארוכה אחת. השיטה ()HTMLEncode של האובייקט Server משמשת למניעת הצגת קוד HTML כלשהו בגוף ההודעה. אם ההודעה אינה מקודדת ל-HTML לפני הצגתה, ואם ההודעה המקורית מכילה תווי HTML מיוחדים, עלולה ההודעה להפוך לבלתי קריאה.

# הוספת הודעה חדשה

כאשר מוסיפים הודעה חדשה מופיע הטופס שבתרשים 27.4. כאשר טופס זה נשלח משמשת שגרת SQL שבתדפיס 27.1 להוספת ההודעה החדשה לבסיס הנתונים. שגרה זו משתמשת בהצהרת SQL בשם INSERT כדי להוסיף הודעה חדשה לטבלת ההודעות. אחר כך, היא יוצרת מספר סודר ייחודי עבור ההודעה החדשה, על ידי מספר הסודר הגבוה ביותר וקידומו ב-1 (אחד). הדבר מבטיח כי לכל הודעה חדשה בפורום יהיה מספר סודר ייחודי.

**תרשים 27.4** טופס כתיבת הודעה חדשה

**תדפיס 27.1** הוספת הודעה חדשה - postMessage

```
create procedure postMessage
(
    @forumName varchar( 30 ),
    @subject varchar( 30 ),
    @username varchar( 30 ),
    @email varchar( 70 ),
    @newMessageID int OUTPUT
)
as
declare @maxOrderNum int

insert messages (
    m_forumName,
    m_subject,
    m_username,    m_email,
    m_reply
```

```
    ) values (
    @forumName,
    @subject,
    @username,
    @email,
    0
    )

select @newMessageID = @@IDENTITY

select @maxOrderNum = max( m_ordernum ) + 1
from messages
where m_forumName = @forumName

update messages set m_ordernum = @maxOrderNum
where m_id = @newMessageID
```

## תגובה להודעה

כאשר אתה מגיב להודעה מופיע הטופס שבתרשים 27.5. תגובה להודעה דומה להוספת הודעה חדשה. ההבדל העיקרי בין השתיים הוא בכך שמספר הסודר המשויך לתגובה זהה לזה של ההודעה המקורית. הדבר מבטיח שהתגובות המתאימות משויכות כהלכה להודעות המקוריות המתאימות. תגובות חדשות נוספות לטבלת בסיס הנתונים באמצעות שגרת SQL המוצגת בתדפיס 27.2.

**תרשים 27.5** תגובה להודעה קיימת

**תדפיס 27.2** הוספת תגובה להודעה - postReply

```
create procedure postReply
(
    @forumName varchar( 30 ),
    @subject varchar( 30 ),
    @username varchar( 30 ),
    @email varchar( 70 ),
    @newMessageID int OUTPUT,
    @reply integer
)
as

insert messages (
    m_forumName,
    m_subject,
    m_username,     m_email,
    m_reply
    ) values (
    @forumName,
    @subject,
    @username,
    @email,
    1
    )

select @newMessageID = @@IDENTITY

select @reply = m_ordernum
from messages
where m_id = @reply

update messages set m_ordernum = @reply
where m_id = @newMessageID
```

# הרחבת פורום הדיון

לפורום הדיון המתואר בפרק זה ניתן להוסיף מספר תכונות יעילות. למשל, זה יכול להיות נחמד אם בכל פעם שמישהו מפרסם תגובה חדשה להודעה, תישלח למחבר ההודעה המקורית הודעת דואר אלקטרוני המכילה את התגובה שפורסמה בפורום, ולכל שאר מפרסמי התגובה להודעה זו.

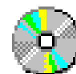
בפרק 10 למדת כיצד להשתמש ב-CDO עבור NTS כדי לשלוח הודעות דואר אלקטרוני מתוך דף ASP. תוכל גם להשתמש ב-CDO עבור NTS כדי לשלוח באופן אוטומטי הודעות דואר אלקטרוני מתוך פורום הדיון. כדי לעשות זאת, השתמש בשגרה sendMail המתוארת בפרק 10.

תכונה יעילה נוספת אותה ניתן להוסיף לפורום הדיון שלך היא הארת תחביר (Syntax Highlighting). כאשר מישהו מפרסם הודעה המכילה תסריט ASP בפורום שבאתר ASPSite, מופיע התסריט באופן אוטומטי עם רקע צהוב (בדיוק כפי שהוא מופיע ב-Visual InterDev של Microsoft). דבר זה מושג על ידי הוספת תגית הסגנון <SPAN> של CSS (Cascading Style Sheet) לגוף ההודעה. להלן הפונקציה המבצעת את ההחלפה:

```
function highlight( byVal theMessage )
    pos =INSTR( theMessage,"&lt;%")
    if pos > 0 then endpos = INSTR( 1, theMessage ,"%&gt;" )
    while ( pos > 0 and endpos > pos )
    theMessage = Replace( theMessage, "&lt;%",
    ↳ "<SPAN CLASS=""yellow"">!lt;%", 1, 1)
    theMessage = Replace( theMessage, "%&gt;", "%!gt;</SPAN>", 1, 1)
    pos = INSTR( theMessage, "&lt;%")
    if pos > 0 then endpos = INSTR(1, theMessage, "%&gt;" )
    wend
    theMessage = Replace( theMessage, "!lt;", "&lt;")
    theMessage = Replace( theMessage, "!gt;", "&gt;")
    highlight = theMessage
end function
```

פונקציה זו מחליפה כל מופע של מחרוזת מקודדת HTML מסוג <% במחרוזת מקודדת HTML הבאה: <SPAN CLASS=""yellow""><%, וכל מופע של המחרוזת %> במחרוזת הזו: %></SPAN>. אם כלל הסגנון הבא מוסף לראש המסמך, כל תסריט הנכלל בין תוחמי תסריט ASP יופיע עם רקע צהוב:

```
<STYLE>
.yello {background: yellow}
</STYLE>
```



דוגמה לדף ASP העושה שימוש בפונקציה זו תוכל למצוא בתקליטור המצורף לספר, בקובץ בשם highlight.asp.

אני בטוח שתמצא בעצמך מספר תכונות יעילות נוספות. אם אתה מעוניין לחלוק אותן עם קוראים נוספים של הספר תוכל להיכנס לפורום שבאתר המלווה את הספר (www.aspsite.com) ולפרסם אותן שם.

# סיכום

בפרק זה למדת כיצד ליצור פורום דיון מושלם באמצעות דפי ASP בלבד. לפורום דיון זה יש תכונות מתקדמות רבות, כגון קישור הודעות וסנכרון מספר מסגרות בו-זמנית.

## פרק 28

# יצירת חנות מקוונת

**בפרק זה:**

- ניווט בחנות המקוונת
- יצירת מיקום האתר
- הכנת מסד הנתונים
- מרכיבים שכיחים
- Global.asa
- דף הבית של החנות המקוונת
- חיפוש אחר פריטים
- הדף העמוס ביותר ביישום
- השלמת ההזמנה
- הזדמנויות לשיפור

אחד מהתחומים שהתפתח חזק ביותר בשנים האחרונות של השימוש באינטרנט הוא תחום המסחר האלקטרוני (eCommerce). גולשי האינטרנט יכולים כיום להזמין מוצרים ממיגוון גדל והולך של חנויות מקוונות. חברות רבות גילו שהן יכולות להציג את מוצריהן למכירה באמצעות אתרים ברשת האינטרנט, ומספר גדל והולך של משתמשים פונים לאינטרנט כדי לרכוש את המוצרים שהם זקוקים להם. ניתן לציין כעובדה, כי מספר חברות מעמידות למכירה מוצרים אותם ניתן לרכוש באמצעות האינטרנט בלבד, ובכך הם חוסכים לעצמם את הצורך בהשכרת שטחי מסחר גדולים ויקרים בהם צריך הלקוח לבקר!

בפרק זה תסקור את השימוש ב-ASP ליצירת חנות מקוונת. במהלך הפרק תיצור אתר היכול לשמש כחנות מקוונת לשימושו של העסק המעוניין למכור את מרכולתו באמצעות האינטרנט.

האתר אותו תיצור בפרק זה יהיה כללי למדי, ופשוט. הוא נועד להוות נקודת פתיחה שתשמש אותך, במידה ותרצה לפתח אתר עסקי עבור הארגון שלך. חנות מקוונת מלאה ושלמה יכולה להיות פשוטה כמו היישום לדוגמה המובא כאן, או מורכבת מספיק כדי לצרוך צוות פיתוח גדול שיעמול שבועות, ואף חודשים, על פיתוחו. פיתוח כגון זה הוא נושאו של ספר שלם אחר.

**הערה:**

אם תרצה לראות דוגמה עובדת של אתר זה, בקר בכתובת:
http://www.aspsite.com/aspstore.
בכתובת זו תמצא את היישום לדוגמה המלא, כפי שהוא מתואר בספר זה.

# ניווט בחנות המקוונת

החנות המקוונת לדוגמה המוצגת כאן מכילה סדרה של דפי ASP שנועדו לעבוד בהרמוניה האחד עם השני. כאשר משתמש מבקר באתר הוא יתחיל את הסיור בדף default.asp, דף הפתיחה של אתר החנות המקוונת שלך. הוא יציג בפני המשתמש את המידע הנחוץ לו לשם שיטוט וניווט מדף זה ואילך.

החל מדף הפתיחה יכול המשתמש לחפש אחר מוצרים באמצעות מילות מפתח. קיימים בו גם קישורים לסל הקניות של המשתמש, כמו גם אזורים יעילים אחרים כגון דף שירות לקוחות (Customer Services) ודף עזרה. דף הפתיחה יכול גם להכיל מוצרים במבצע שהצגתם תלויה בשעה בה נכנס המשתמש לאתר, מה שיאפשר למשתמש להוסיף לסל הקניות שלו מוצרים, ישירות מדף הפתיחה.

# יצירת מיקום האתר

הגדרת הסביבה שתכיל את אתר החנות המקוונת היא פעולה פשוטה למדי. בשרת האינטרנט שלך, צור תיקיה וירטואלית (Virtual Directory) שתכיל את כל דפי ה-ASP של האתר, כמו גם מספר דפי תמיכה. שם התיקיה הווירטואלית של היישום לדוגמה יהיה ASPStore. בשל כך, הגישה לאתר תתבצע על ידי גישה לכתובת http://servername/aspstore. מכיון שזה יישום ASP העומד בפני עצמו, תצטרך ליצור גם יישום אינטרנט (Web Application) בשם ASPStore.

**הערה:**

כדי ללמוד ביתר הרחבה אודות יצירת יישומי ASP, פנה ל**פרק 4**.

**טיפ:**

אם תרצה להקים חנות מקוונת לדוגמה בשרת האינטרנט שלך, תמצא את כל הקבצים הנדרשים לשם כך בתקליטור המצורף לספר זה. כדי להגדיר את החנות, עקוב אחר ההוראות שבקובץ setup.htm שבתיקיה של פרק זה בתקליטור.

# הכנת מסד הנתונים

יישום החנות המקוונת שלך ישמש כתעלה דו-כיוונית למידע העובר אל ומאת משתמש הקצה. הפונקציה הראשונית של האתר תהיה להציג בפני המשתמש מידע לגבי המוצרים שלך. המשתמש, בתגובה, ישלח לאתר מידע לגבי מוצרים אותם הוא מבקש לרכוש.

לאתר לדוגמה שלך תיצור מסד נתונים יחסי המכיל שלוש טבלאות:

- הטבלה Products אשר תאחסן את כל הנתונים אודות המוצרים שאתה מוכר.
- הטבלה Orders אשר בה תאוחסן רשומה עבור כל הזמנה שבוצעה על ידי לקוח.
- הטבלה Order Details אשר תכיל רשומה עבור כל פריט בכל הזמנה. רשומות בטבלה זו מייצגות פריטי שורה (Line Items) המשויכות לרשומות הזמנה בטבלה Orders.

בשל העובדה שהוא קיים כמעט בכל מחשב עסקי, נבחר היישום Access של Microsoft להוות את הפלטפורמה להדגמה זו. אבל, כפי שתראה הגישה למסד הנתונים מתבצעת תוך שימוש ב-ADO (ActiveX Data Objects) דרך ODBC, כך שפלטפורמת מסד הנתונים עצמה אינה רלוונטית ליישום. אתה תראה כיצד ליצור חיבור כללי ("generic" connection) למסד הנתונים, אשר יכול להשתנות בקלות כדי לפעול עם פלטפורמות שונות.

**אזהרה:**

למרות שהשימוש ב- Access מספק לצורך הפיתוח, הוא אינו מתאים לאתר אינטרנט אף אם תהיה בו תנועה מתונה. לפני שתעביר את יישום החנות המקוונת שלך לסביבת עבודה עסקית, עליך לשקול את השימוש במנוע מסד נתונים הפועל בסביבת שרת/לקוח (כגון שרת SQL של Microsoft).

# טבלת Products

הטבלה Products מאחסנת את השם, תיאור, מחיר ומידע נוסף אודות המוצרים העומדים למכירה בחנות שלך. היא מכילה את השדות הבאים:

- ItemID - שדה מספרי המכיל מספר שלם ארוך (Long Integer), המשמש כמזהה רשומה.
- ItemProductNumber - שדה טקסט המכיל מזהה כגון מק״ט או קוד UPC אשר הוא ייחודי למוצר.
- ItemName - שדה טקסט המכיל את שם המוצר.
- ItemDescription - שדה טקסט ארוך יותר המאפשר תיאור מפורט יותר של המוצר.
- ItemPictureFile - שמו של הקובץ הגרפי שישמש לאחזור תמונה של המוצר להצגה בדפי האינטרנט. הקבצים הגרפיים אמורים להיות מאוחסנים בתיקיה images מתחת לתיקיה products, שהיא תיקיית משנה בתיקיית היישום שלך.
- ItemRegularPrice - שדה מטבע (Currency) המכיל את מחירו הרגיל של המוצר.
- ItemSlaePrice - אם למוצר יש כרגע מחיר מבצע או מחיר מיוחד, הוא יופיע בשדה זה. אחרת, שדה זה יכיל את הערך 0 (אפס).
- BeginSpecial - שדה תאריך/שעה (Date/Time) המכיל את התאריך הראשון בו המוצר אמור להיות מוצג בחלק Today's Featured Product (המוצר החם של היום) שבדף הראשי של החנות המקוונת.
- EndSpecial - שדה תאריך/שעה המכיל את התאריך האחרון בו המוצר אמור להיות מוצג בחלק Today’s Featured Product (המוצר החם של היום) שבדף הראשי של החנות המקוונת.

# טבלת Orders

לטבלה זו נוספת רשומה בכל פעם שלקוח מבצע הזמנה של מוצר. טבלה זו תכיל את השדות הבאים:

- OrderID - שדה מספרי המכיל ערך שלם ארוך מהקודם באופן אוטומטי ואשר משמש כמזהה רשומה.

- OrderDate - שדה תאריך/שעה המכיל את התאריך בו הושלמה ההזמנה.
- OrderTime - שדה תאריך/שעה המכיל את השעה בה הושלמה ההזמנה.
- CustomerName - שדה טקסט המכיל את שמו של המשתמש שביצע הזמנה זו.
- CustomerAddress - שדה טקסט המכיל את הכתובת של המשתמש שביצע הזמנה זו.
- CustomerCity - שדה טקסט המכיל את העיר של המשתמש שביצע הזמנה זו.
- CustomerState - שדה טקסט המכיל את המדינה של המשתמש שביצע הזמנה זו.
- CustomerZip - שדה טקסט המכיל את המיקוד של המשתמש שביצע הזמנה זו.
- CustomerPhone - שדה טקסט המכיל את מספר הטלפון של המשתמש שביצע הזמנה זו.
- OrderAmount - שדה מטבע המכיל את סך כל ההזמנה.
- PaymentMethod - שדה טקסט המכיל את שם כרטיס האשראי המשמש לתשלום עבור ההזמנה.
- PaymentAccount - שדה טקסט המכיל את מספר כרטיס האשראי המשמש לתשלום עבור ההזמנה.
- ExpDate - שדה טקסט המכיל את תאריך תוקף כרטיס האשראי המשמש לתשלום עבור ההזמנה.

## טבלת Order Details

בכל פעם שמבוצעת הזמנה, מתווספת רשומה נוספת לטבלה Orders; בנוסף, לטבלה Order Details מתווספת רשומה נוספת עבור כל פריט שבהזמנה זו. דבר זה מגדיר יחסי אחד-לרבים אפקטיביים בין שתי הטבלאות.

- OrderID - שדה מספרי זה מכיל מספר שלם ארוך המזהה את הרשומה מהטבלה Orders אליה מתייחסת רשומת פירוט זו.
- ProductNumber - שדה טקסט זה מכיל את מספר המוצר (מהטבלה Products) המזהה את המוצר המיוצג על ידי רשומת פירוט זו.
- Quantity - שדה מספרי המייצג את כמות הפריטים שהוזמנה מהמוצר.
- Price - שדה מטבע המכיל את המחיר (הרגיל, מבצע או מיוחד) של המוצר הנרכש, בעת ביצוע ההזמנה.

## אחסון הנתונים

לאחר שיצרת את מסד הנתונים צריך שרת האינטרנט ליצור עימו קשר באמצעות ODBC connection. לצורך אתר הדוגמה, הגדר DSN (System Data Source Name) של ASPStore; ODBC connection זה ישמש לצורך התחברות למסד הנתונים. בסעיף הבא, Global.asa, חיבורי הנתונים (data connection) של החנות המקוונת מאוחסנים כמשתני מערכת (System Variables) בכל פעם שמתחיל Session.

**הערה:**

למידע נוסף אודות יצירת חיבורי (data connection) נתונים למסדי נתונים ראה פרק 12.

# מרכיבים שכיחים

כדי להבטיח מראה ותחושה של אחידות בכל היישום, עיצבתי שני קבצי INCLUDE אשר יתנו לכל דף באתר כותרת עליונה וכותרת תחתונה אחידות. השימוש בקבצי INCLUDE אלה יקל על ביצוע שינויים לחלקם העליון והתחתון של כל הדפים באתר.

## HEADER.INC

הקובץ HEADER.INC, המוצג בתדפיס 28.1, נכלל קרוב לחלקו העליון של כל דף, איפשהו בין התגיות <HTML>. אבל, לפני שמיושמת תגית Include של HTML, קובע קוד ASP בכל דף את הערכים של שני משתנים בהם ישתמש HEADER.INC. sPageHeader משמש לקביעת כותרת הדף בתוך התגית <HEAD>. SPageCaption מציין את כותרת הדף שתופיע בסרגל המטושטש (Blur) הדומה לשורת התאריך בכותרת עיתון, מתחת לכל לוגו בכל דף.

בפינה השמאלית-עליונה של כל דף, מציב HEADER.INC לוגו של האתר; לוגו זה מתפקד כקישור <A HREF> לדף default.asp, הדף הראשי של היישום. בנוסף קיימים גם קישורי ניווט בפינה הימנית-עליונה, לשני דפים אליהם המשתמשים ניגשים לעיתים קרובות.

במקרים רבים נחליט להגדיר באתרנו מספר קבצים שיהיו מוכללים באמצעות Include וזאת כדי למנוע הקלדת קוד כל פעם מחדש.

**תדפיס 28.1** HEADER.INC

```
<HEAD>
<TITLE><% = sPageHead %></TITLE>
</HEAD>

<BODY>
<TABLE BORDER="0" WIDTH="100%">
  <TR>
    <TD WIDTH="70%"><A HREF="default.asp">
    <IMG SRC="images/OnlineStore.gif"
     ALT="ASP Unleashed Online Store Home" BORDER="0"></A></TD>
    <TD VALIGN="top">
      <TABLE BGCOLOR="#dc143c" BORDER="0" WIDTH="100%">
        <TR>
          <TD BGCOLOR="#87cefa" ALIGN="center">
          <FONT FACE="Verdana,Arial" SIZE=-1>
          <A HREF="default.asp">Online Store Home</A>
          </FONT></TD>
        </TR>
        <TR>
          <TD BGCOLOR="#87cefa" ALIGN="center">
          <FONT FACE="Verdana,Arial" SIZE=-1>
          <A HREF="shopcart.asp">Shopping Cart</A>
          </FONT></TD>
        </TR>
      </TABLE>
    </TD>
  </TR>
</TABLE>
<TABLE BORDER="0" WIDTH="100%">
  <TR>
    <TD BGCOLOR="#0000FF" ALIGN="center">
    <FONT FACE="Verdana,Arial" COLOR="#ffff00">
    <B><% = sPageCaption %></FONT></B></TD>
  </TR>
</TABLE>
```

# FOOTER.INC

בדומה ל-HEADER.INC, FOOTER.INC נועד לתת לכל חלק תחתון של כל דף באתר מראה דומה. הוא מכיל סדרה של קישורים יעילים ולאחריהם נתוני שמירת זכויות היוצרים של האתר. הפניית ה-INCLUDE עבור FOOTER.INC צריכה להופיע מייד לפני תגיות הסגירה </BODY> ו- </HTML> בכל דף. הקוד של FOOTER.INC מופיע בתדפיס 28.2.

**תדפיס 28.2** FOOTER.INC

```
<BR><TABLE BORDER="0" WIDTH="100%">
 <TR ALIGN="center">
  <TD BGCOLOR="#33ccff">
   <FONT FACE="Verdana, Arial" COLOR="#ffff33" SIZE="-1">
   <A HREF="help.asp">Help</A></FONT></TD>
  <TD BGCOLOR="#33ccff">
   <FONT FACE="Verdana, Arial" COLOR="#ffff33" SIZE="-1">
   <A HREF="shopcart.asp">Shopping Cart</A></FONT></TD>
  <TD BGCOLOR="#33ccff">
   <FONT FACE="Verdana, Arial" COLOR="#ffff33" SIZE="-1">
   <A HREF="custsvc.asp">Customer Service</A></FONT></TD>
  <TD BGCOLOR="#33ccff">
   <FONT FACE="Verdana, Arial" COLOR="#ffff33" SIZE="-1">
   <A HREF="feedback.asp">Feedback</A></FONT></TD>
 </TR>
</TABLE>
<BR>
<HR ALIGN="CENTER" SIZE="2" COLOR="#0000FF">
<DIV ALIGN="center"><FONT FACE="Arial" COLOR="#808080" SIZE=-2>
The ASP Unleashed Online Store web site was created by
<A HREF="mailto:jspotts@bigfoot.com">Jeff Spotts</A>.
Copyright &copy; 1999 Jeff Spotts. All Rights Reserved.
</FONT>
</DIV><HR ALIGN="CENTER" SIZE="2" COLOR="#0000FF">
```

# Global.asa

יצרתי קובץ Global.asa, כפי שמוצג בתדפיס 28.3, אשר מגדיר משתני session שישמשו בכל היישום. משימה זו מבוצעת על ידי הצבת קוד בשגרה Session_OnStart שבקובץ Global.asa ; קוד זה מופעל בכל פעם שמשתמש יוצר session עם השרת.

**הערה:**

למידע מפורט יותר אודות השימוש בקובץ Global.asa ומשתני session, ראה בפרק 4 ובפרק 5.

**תדפיס 28.3** Global.asa

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
Sub Session_OnStart
 dim sItemNums(0), nItemQtys(0)
 sItemNums(0) = ""
 nItemQtys(0) = 0
```

```
  session("CartItemNumArray") = sItemNums
  session("CartItemQtyArray") = nItemQtys
  session("DBLogin") = "DSN=ASPStore"
  session("shipbasecharge") = 4.50
  session("shipitemcharge") = 1.25

End Sub
</SCRIPT>
```

הקובץ Global.asa יוצר את המשתנים הבאים עבור כל session:

- Session("CartItemNumArray") - זהו מערך שיכיל מספרי פריטים המזהים את המוצרים אותם מוסיף המשתמש לסל הקניות שלו.
- Session("CartItemQtyArray") - מערך נוסף אשר יכיל את הכמויות של המוצרים התואמים אותם מוסיף המשתמש לסל הקניות שלו.
- Session("DBLogin") - מכיל את נתוני הכניסה למערכת הנדרשים לשם התחברות למקור הנתונים העיקרי של היישום. בהנחה שנוצר ODBC connection בשרת, ה-DSN יהיה חלק ממשתנה זה.
- Session("shipbasecharge") ו- Session("shipitemcharge") - מכילים את הוצאות המשלוח עבור הזמנות המבוצעות באתר האינטרנט שלך. המשתנה הראשון מייצג את החיוב לפי הזמנה (Per-Order); המשתנה השני מייצג את החיוב לפי פריט (Per-Item). זאת אומרת, סך כל מחיר המשלוח לכל הזמנה (TC) יהיה: חיוב בסיסי (BC) ועוד חיוב לפי פריט (PI) כפול מספר פריטים (NI). המשוואה היא:

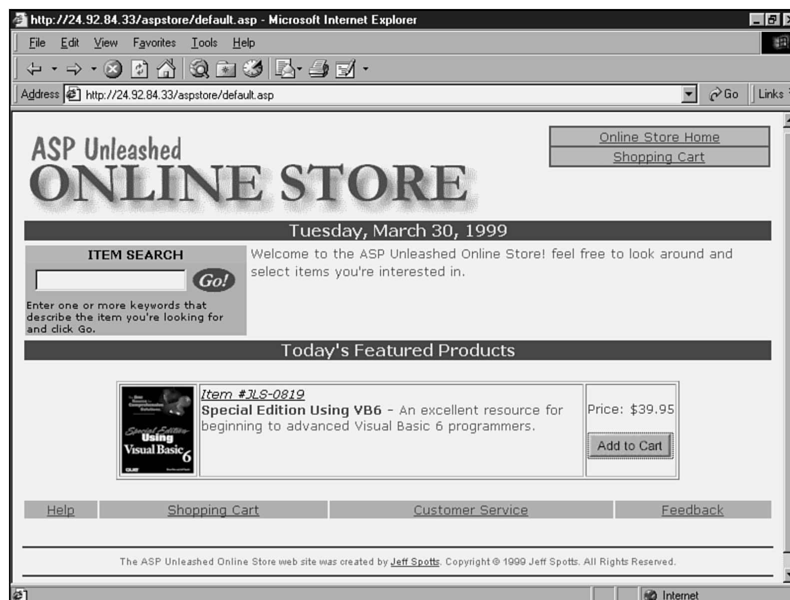
BC + (PI x NI) = TC

שני המשתנים Session("CartItemNumArray") ו-Session("CartItemQtyArray"), ישמשו להעברת תוכן סל הקניות של המשתמש מדף לדף. שלושת המשתנים האחרים, Session("DBLogin"), Session("shipbasecharge") ו-Session("shipitemcharge"), נוצרים כאן כך שניתן יהיה לשנות אותם בקלות, במידת הצורך.

# דף הבית של החנות המקוונת

דף הפתיחה של האתר שלך, default.asp הוא הדבר הראשון הנגלה לעיניהם של המשתמשים כאשר הם נכנסים לאתר. בשל כך, חייב דף הבית (דף הפתיחה, עמוד ראשי, שער) לתפוס את העין ביופיו, ועדיין להיות פשוט דיו כדי שלא להמם את המשתמש. הדף default.asp מוצג בתרשים 28.1.

הקונים יוכלו להתחיל מיד במסע הקניות שלהם, באחת משתי הדרכים, מייד לאחר שיוצג בפניהם דף הפתיחה. דרך אחת, באמצעות תיבת טקסט בצד שמאל המאפשרת למשתמש להקליד מילות מפתח ולבצע חיפוש אחר מוצר. דרך שנייה, באמצעות רשימת פריטים הנמכרים כעת במחיר מבצע ומוצגים בתחתית הדף; לחיצה על הכפתור Add to Cart המופיע ליד כל מוצר מאפשר למשתמש לבחור במוצר בקלות ובמהירות.




**תרשים 28.1** דף הפתיחה של החנות המקוונת צריך להיות מאוד ידידותי למשתמשים בו

# Default.asp

חוץ מהקבצים HEADER ו-FOOTER שהוזכרו קודם לכן, הקובץ default.asp גם מכיל שני קבצי INCLUDE נוספים:

- ADOVBS.INC - מסופק על ידי Microsoft, מאפשר ליישומי VBScript גישה קלה לקבועים מוגדרים מראש של ADO.
- PROCS.INC - מכיל הליך ופונקציה אשר ייעשה בהן שימוש במספר דפים ביישום. על ידי אחסון קוד זה בקובץ INCLUDE המיובא לתוך דפים כפי שנדרש, ניתן להפחית את שורות הקוד שיש להקליד כל פעם מחדש (או להעתיק).

שתי התכונות המעניינות בקובץ default.asp, מנקודת מבט של מפתח האתר, הם ה-<FORM> המאפשר למשתמש להקליד מילות מפתח, ואזור קוד ASP הבוחר ומציג את מוצר היום הנבחר ממסד הנתונים.

**תדפיס 28.4** Default.asp

```
<!--#INCLUDE FILE="adovbs.inc"-->
<!--#INCLUDE FILE="procs.inc"-->

<%
   sPageHead = "ASP Unleashed Online Store Home Page"
   sPageCaption = formatdatetime(now,vblongdate)
```

```
    session("redirectto") = "default.asp"
    session("salecomplete") = false
%>
<HTML>
<!--#INCLUDE FILE="header.inc"-->

<TABLE BORDER="0" WIDTH="100%">
  <TR>
    <TD WIDTH="30%">
      <TABLE BORDER="0" WIDTH="100%" BGCOLOR="#c0c0c0" CELLSPACING="0">
        <TR>
          <TD ALIGN="center" BGCOLOR=#66ccff>
          <FONT FACE="Verdana,Arial" SIZE=-1>
          <STRONG>ITEM SEARCH</STRONG></FONT></TD>
        </TR>
        <TR>
          <TD ALIGN="center">
            <TABLE BORDER="0">
              <TR>
                <FORM ACTION="prodsearch.asp" METHOD="POST">
                <TD><INPUT TYPE="Text" NAME="txtSearch" SIZE="20"
                  maxlength="100"></TD>
                <TD><INPUT TYPE="image" SRC="images/go.gif" BORDER="0"
                  ALT="Click to begin search"></TD>
                </FORM>
              </TR>
            </TABLE>
          </TD>
        </TR>
        <TR>
          <TD><FONT FACE="Verdana,Arial" COLOR="#000000" SIZE="-2">
          Enter one or more keywords that describe the item you're
          looking for and click Go.</FONT></TD>
        </TR>
      </TABLE>
    </TD>
    <TD VALIGN="top">
      <FONT FACE="Verdana,Arial" COLOR="#008000" SIZE="-1">
      Welcome to the ASP Unleashed Online Store! feel free to look around
      and select items you're interested in.</FONT>
    </TD>
  </TR>
</TABLE>

<%
  set dbMain = server.createobject("ADODB.Connection")
  dbMain.open session("dblogin")
  Set rsProducts = Server.CreateObject("ADODB.Recordset")
```

```
  sSQL = "SELECT * FROM products WHERE beginspecial <= #" & date() &
    ↳ "#" & " AND endspecial >= #" & date() & "#"
  rsProducts.open sSQL, dbMain, adopenstatic, adlockpessimistic, adcmdtext
%>
<% if not rsproducts.eof then %>
<TABLE BORDER="0" WIDTH="100%">
  <TR>
  <TD BGCOLOR="#0000FF"><FONT FACE="Verdana,Arial" COLOR="#ffff00">
    <P ALIGN="center"><B>Today's Featured Products</B></P></FONT></TD>
  </TR>
</TABLE>
<BR>
<TABLE WIDTH="75%" BORDER="1" ALIGN="center">
<%
  rsproducts.movefirst
  do while not rsproducts.eof
    WriteProductRow
    rsProducts.movenext
  loop
%>
</TABLE>
<% end if %>
<!--#INCLUDE FILE="footer.inc"-->
</BODY>
</HTML>
```

# הכנת חיפוש

קטע הטופס (<FORM>) הנכלל בדף default.asp מכיל תיבת קלט לטקסט, txtSearch, המאפשרת למשתמש להקליד בה מילות מפתח התואמות לשם, מספר פריט או תיאורו, בניסיון לאתר את המוצר. תמונת לחצן Go שולחת את הטופס ליעדו, Prodsearch.asp, אשר בתורו מבצע את החיפוש עצמו, ומציג את תוצאותיו. מאוחר יותר בהמשך הפרק נדון ב-Prodsearch.asp עצמו.

# הפעלת החיפוש

בשורות האחרונות של default.asp נבנית הצהרת SQL המחפשת אחר רשומות בטבלה Products של מסד הנתונים הראשי, התאריך הנוכחי בה הוא איפה שהוא בין ערך השדה beginspecial לבין ערך השדה endspecial. הצהרת SQL זו, בתורה, משמשת ליצירת אובייקט Recordset בשם rsProducts, המכיל רשומות העונות לקריטריונים אלה. המאפיין EOF של אובייקט Recordset נבדק כדי לקבוע האם נמצאו רשומות העונות על הקריטריונים. אם כן, נוצרת טבלה כדי להציג את המוצרים.

לאחר שהטבלה נוצרת מופעלות שורות הקוד הבאות :

```
rsproducts.movefirst
  do while not rsproducts.eof
    WriteProductRow
    rsProducts.movenext
  loop
```

קוד זה יוצר לולאה על הרשומות ב-Recordset בשם rsProducts, תוך קריאה לשגרה בשם WriteProductRow בכל פעם מחדש. WriteProductRow אינה נראית בקובץ default.asp. היא נכללת בקובץ PROCS.INC, כפי שתואר קודם לכן.

# הצגת המוצר

המטרה של WriteProductRow, כפי שמוצג בתדפיס 28.5, היא ליצור בטבלה מוגדרת מראש שורה אחת המציגה את המוצר המסוים המיוצג על ידי הרשומה הנוכחית של Recordset בשם rsProducts. שורה זו מכילה את התאים הבאים :

- תמונת המוצר, אם קיימת. תמונה זו נוצרת על ידי הפניית שם קובץ תמונה המאוחזר מהשדה ItemPictureFile שבטבלה Products.
- תא מידע המכיל את מספר הפריט, שם ותיאור. כל אחד ממרכיבים אלה מעוצב בגופן שונה, כדי להקל על זיהויו.
- תא תמחור המציג את מחירו הנוכחי של המוצר. אם למוצר יש מחיר מבצע (המושג מהשדה ItemSalePrice) מדווח גם המחיר הרגיל, למטרות השוואה. תא זה גם מכיל קישור לקובץ shopcart.asp המכיל מחרוזת שאילתה המורה ל-shopcart.asp להיות מופעל במצב הוספה (Add Mode), ומחרוזת שאילתה נוספת המעבירה את מספר הפריט.

**תדפיס 28.5** PROCS.INC

```
<%
'PROCS.INC
Sub WriteProductRow
  'Write a table row for the current record of rsProducts
  response.write "  <TR>" & vbcrlf

  'First cell for picture
  sTemp = trim("" & rsproducts("itempicturefile"))
  if sTemp = "" then
    sTemp = "Picture Not Available"
  else
    sTemp = "<img src='images\products\" & sTemp & "' BORDER=0 alt=''>"
  end if
  response.write "    <TD ALIGN='center'>"
  response.write "<font face='Verdana, Arial' COLOR='#008000' SIZE='-1'>"
  response.write sTemp & "</FONT></TD>" & vbcrlf
```

```
  'Second cell for Item #, name, description
  response.write "    <TD VALIGN='top'>"
  response.write "<font face='Verdana, Aria' COLOR='#000000' SIZE='-1'>"
  response.write "<i><u>Item #" & rsProducts("itemproductnumber")
  response.write "</u></i></FONT><BR>"
  response.write "<font face='Verdana, Arial' COLOR='#0000ff' SIZE='-1'>"
  response.write "<B>" & rsProducts("itemname") & "</B> - "
  response.write "<font face='Verdana, Arial' COLOR='#008000' SIZE='-1'>"
  response.write rsProducts("itemdescription") & "<BR><BR>"
  response.write "</FONT></TD>" & vbcrlf

  'Third cell for price and "add to cart" link
  if rsproducts("itemsaleprice") > 0 then
    sTemp = "SALE: " & formatcurrency(rsProducts("itemsaleprice")) &
      ⇘  "<BR>(Regular " &
      ⇘  formatcurrency(rsProducts("itemregularprice")) & ")"
  else
    sTemp = "Price: " & formatcurrency(rsProducts("itemregularprice"))
  end if
  response.write "    <TD ALIGN='center' valign='middle' nowrap>"
  response.write "<font face='Verdana, Arial' COLOR='#0000ff' SIZE='-1'>"
  response.write sTemp & "<BR><BR>"
  response.write "<a href='shopcart.asp?mode=add&itemno="
  response.write rsproducts("itemproductnumber") & "'></FONT>"
  response.write "<img src='images/addtocart.jpg' BORDER='0'></a></TD>"
  response.write vbcrlf

  response.write "  </TR>"  & vbcrlf

End Sub

Function RemoveQuotes(s)
  dim i, s2
  s = trim(s)

  if len(s) = 0 then
    RemoveQuotes = ""
    exit function
  end if
```

```
s2 = ""
  for i = 1 to len(s)
    if mid(s,i,1) = "'" then
      s2 = s2 & "''"
    else
      s2 = s2 & mid(s, i, 1)
    end if
  next
  RemoveQuotes = s2
End Function
%>
```

Shopcart.asp הוא דף בו המוצר נוסף לסל הקניות של המשתמש. נדון בו ביתר הרחבה בסעיף, **הדף העמוס ביותר ביישום**, שבהמשך פרק זה.

# חיפוש אחר פריטים

אם המשתמש מקליד מידע לתיבת טקסט בדף default.asp ולוחץ על לחצן Go, יעד הטופס prodsearch.asp) המוצג בתדפיס 28.6) נטען ומופעל. דף זה אחראי על ביצוע חיפוש אחר מוצר, בהתבסס על נתוני מילות המפתח אותן הקליד המשתמש בעת שצפה בדף default.asp. אם אותר מוצר אחד או יותר, מוצג כל אחד מהם באמצעות השגרה WriteProductRow המאוחסנת בקובץ PROCS.INC, בו דן הסעיף הקודם. התוצאה של חיפוש טיפוסי מתוארת בתרשים 28.2.

**תרשים 28.2** תוצאות החיפוש המוצגות בפני משתמש שערך חיפוש מילות מפתח

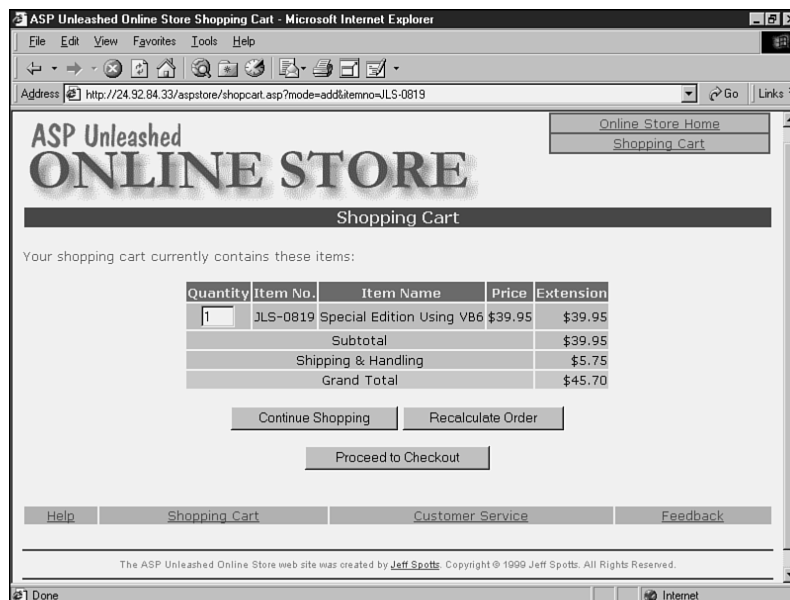
```
<!--#INCLUDE FILE="adovbs.inc"-->
<!--#INCLUDE FILE="procs.inc"-->
<%
    sPageHead = "ASP Unleashed Online Store Product Search Results"
    sPageCaption = "Product Search Results"


%>
<HTML>

<!--#INCLUDE FILE="header.inc"-->

<BR><FONT FACE="Verdana,Arial" COLOR="#008000" SIZE="-1">
You searched for products containing the text
"<% = request.form("txtSearch")%>"
in their names, item numbers, or descriptions.
<BR><BR>
<%
  set dbMain = server.createobject("ADODB.Connection")
  dbMain.open session("dblogin")
  Set rsProducts = Server.CreateObject("ADODB.Recordset")
  sSearch = RemoveQuotes(request.form("txtSearch"))
  sSQL = "SELECT * FROM products WHERE itemproductnumber LIKE '%" &
    ↳ sSearch & "%' OR itemname LIKE '%" &
    ↳ sSearch & "%' OR itemdescription LIKE '%" &
    ↳ sSearch & "%'"

  rsProducts.open sSQL, dbMain, adopenstatic, adlockpessimistic, adcmdtext

  if rsproducts.eof then
    response.write "No products matching that criteria were found."
    response.write vbcrlf
  else
    response.write "<TABLE WIDTH='75%' BORDER='1' ALIGN='center'>"
    response.write vbcrlf
    rsproducts.movefirst
    do while not rsproducts.eof
      WriteProductRow
      rsProducts.movenext
    loop
    response.write "</TABLE>" & vbcrlf
  end if
%>

<!--#INCLUDE FILE="footer.inc"-->
</BODY>
</HTML>
```

שים לב שקוד זה קורא לפונקציה בשם RemoveQuotes. פונקציה זו, הנכללת בקובץ PROCS.INC המוצג בתדפיס 28.5, משמשת לשינוי מחרוזות טקסט המוקלדות על ידי המשתמש. היא מחליפה כל סימן גרש בודד בשני סימני גרש בודדים. דבר זה מאפשר למשתמש להקליד קריטריון חיפוש המכיל בעצמו סימני גרש, מבלי לגרום לבעיות למשפטי WHERE של SQL (כאשר אנו מחפשים מחרוזת ב-SQL אנו משתמשים בגרשיים מסביב למחרוזת זו).

לאחר בניית הצהרת SQL המכילה את קריטריוני החיפוש של המשתמש, נוצר Recordset המכיל את הרשומות התואמות. אם המאפיין EOF של רשומה מוגדר לערך True, מופיעה הודעה המורה למשתמש כי לא נמצאו מוצרים העונים על קריטריון החיפוש שהגדיר. אבל, אם קיימות רשומות תואמות, מוצגת בפני המשתמש טבלה המכילה את כל אותן רשומות תואמות. כפי שראית כאשר דנו בדף default.asp, לכל מוצר יש קישור לדף shopcart.asp. בנוסף, שורת הטבלה עבור כל מוצר נכתבת באמצעות השגרה WriteProductRow, הנכללת בקובץ PROCS.INC.

# הדף העמוס ביותר ביישום

הקובץ shopcart.asp, המוצג בתדפיס 28.7, הוא סוס העבודה של יישום החנות המקוונת שלך. כפי שתראה כאשר תבחן את קוד ASP שלו, shopcart.asp נועד לפעול באחד מתוך מספר מצבים; כל אחד מהמצבים מתואר בסעיפים הבאים. תרשים 28.3 מציג את shopcart.asp בפעולה.




**תרשים 28.3** shopcart.asp במצב Add - כדי להוסיף פריט לסל הקניות של המשתמש

```
<!--#INCLUDE FILE="adovbs.inc"-->
<%
 sCartItemNums = session("cartitemnumarray")
 nCartItemQtys = session("cartitemqtyarray")
 nNumCartItems = ubound(sCartItemNums)

 'Determine the mode of this page's operation. If we got here from a
 '  link that had a "mode" querystring of "add", mode is add:
 sMode = request.querystring("mode")

 'Or, we could have gotten here from a form:
 select case request.form("cmdSubmit")
   case "Proceed to Checkout"
     sMode = "checkout"
   case "Recalculate Order"
     sMode = "recalc"
   case "Continue Shopping"
     sMode = "continue"
 end select

 if sMode = "add" then
   nThisItemIndex = 0
   sItemNo = ucase(request.querystring("itemno"))
   if nNumCartItems > 0 then
     for i = 1 to nNumCartItems
       if ucase(scartitemnums(i)) = sItemNo then
         nThisItemIndex = i
         exit for
       end if
     next
   end if
   if nThisItemIndex = 0 then
     nThisItemIndex = nNumCartItems + 1
     redim Preserve sCartItemNums(nThisItemIndex)
     redim Preserve nCartItemQtys(nThisItemIndex)
     sCartItemNums(nThisItemIndex) = sItemNo
     nCartItemQtys(nThisItemIndex) = 1
   else
     nCartItemQtys(nThisItemIndex) = nCartItemQtys(nThisItemIndex) + 1
   end if
   session("cartitemnumarray") = sCartItemNums
   session("cartitemqtyarray") = nCartItemQtys
   nNumCartItems = ubound(sCartItemNums)
 end if
```

```
  if sMode = "recalc" or sMode = "checkout" or sMode = "continue" then
    for i = 1 to nNumCartItems
      nThisQty = int(request.form("txtItemQty" & i))
      nCartItemQtys(i) = nThisQty
    next
    session("cartitemnumarray") = sCartItemNums
    session("cartitemqtyarray") = nCartItemQtys
  end if

  'Move to another page based on mode.
  '  If mode is "add" or "recalc", we stay here.
  '  If mode is "continue", we move back to the main page so the user
  '     can shop more.
  '  If mode is "checkout", we stay here but change the mode th
  select case sMode
    case "continue"
      response.redirect("default.asp")
  end select

  set dbMain = server.createobject("ADODB.Connection")
  dbMain.open session("dblogin")

  if sMode = "checkout" then
    sPageHead = "ASP Unleashed Online Store Checkout"
    sPageCaption = "Checkout"
  else
    sPageHead = "ASP Unleashed Online Store Shopping Cart"
    sPageCaption = "Shopping Cart"
  end if

%>
<HTML>

<!--#INCLUDE FILE="header.inc"-->

<BR>
<FONT FACE="Verdana,Arial" COLOR="#008000" SIZE="-1">
<% if sMode = "checkout" then %>
<FORM ACTION="submitorder.asp" METHOD="POST">
<% else %>
<FORM ACTION="shopcart.asp" METHOD="POST">
<% end if %>
<%
  if nNumCartItems = 0 then
    response.write "There are no items in your shopping cart.<BR>"
  else
    if sMode = "checkout" then
%>
```

```
Your order is ready to submit. Please provide the following information
and click the "Submit Order" button.
<BR><BR>
<TABLE ALIGN="center" BORDER="0">
  <TR BGCOLOR="#c0c0c0">
    <TH COLSPAN="2" ALIGN="center"><FONT FACE="Verdana,Arial" SIZE="-1">
      Customer Information</TD>
  </TR>
  <TR>
    <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">Name:</TD>
    <TD><FONT FACE="Verdana,Arial" SIZE="-1">
      <INPUT TYPE="Text" NAME="txtCustomerName" SIZE="30" MAXLENGTH="30">
      </TD>
  </TR>
  <TR>
    <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">Address:</TD>
    <TD><FONT FACE="Verdana,Arial" SIZE="-1">
      <INPUT TYPE="Text" NAME="txtAddress" SIZE="30" MAXLENGTH="30"></TD>
  </TR>
  <TR>
    <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
      City/State/ZIP:</TD>
    <TD><FONT FACE="Verdana,Arial" SIZE="-1">
      <INPUT TYPE="Text" NAME="txtCity" SIZE="25" MAXLENGTH="30">
      <INPUT TYPE="Text" NAME="txtState" SIZE="4" MAXLENGTH="4">
      <INPUT TYPE="Text" NAME="txtZIP" SIZE="7" MAXLENGTH="7"></TD>
  </TR>
  <TR>
    <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
      Daytime Phone:</TD>
    <TD><FONT FACE="Verdana,Arial" SIZE="-1">
      <INPUT TYPE="Text" NAME="txtPhone" SIZE="15" MAXLENGTH="15"></TD>
  </TR>
  <TR>
    <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
      Payment Method:</TD>
    <TD><FONT FACE="Verdana,Arial" SIZE="-1">
      <INPUT TYPE="Radio" NAME="optPaymentType" VALUE="visa">
        Visa 
      <INPUT TYPE="Radio" NAME="optPaymentType" VALUE="mc">
        Mastercard 
      <INPUT TYPE="Radio" NAME="optPaymentType" VALUE="amex">
        American Express
    </TD>
  </TR>
  <TR>
    <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
      Account Number:</TD>
```

```
    <TD><FONT FACE="Verdana,Arial" SIZE="-1">
      <INPUT TYPE="Text" NAME="txtAcctNo" SIZE="20" MAXLENGTH="20"></TD>
  </TR>
  <TR>
    <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
      Expiration (mm/yyyy):</TD>
    <TD><FONT FACE="Verdana,Arial" SIZE="-1">
      <INPUT TYPE="Text" NAME="txtExpDate" SIZE="10" MAXLENGTH="7"></TD>
  </TR>
</TABLE>
<%
    else
      response.write "Your shopping cart currently contains these items:"
    end if
%>
<BR><BR>
<TABLE ALIGN="center" BORDER="0">
  <TR BGCOLOR="#ff0066">
    <TH><FONT FACE="Verdana,Arial" COLOR="#ffff33" SIZE="-1">
      Quantity</FONT></TD>
    <TH><FONT FACE="Verdana,Arial" COLOR="#ffff33" SIZE="-1">
      Item No.</FONT></TD>
    <TH><FONT FACE="Verdana,Arial" COLOR="#ffff33" SIZE="-1">
      Item Name</FONT></TD>
    <TH><FONT FACE="Verdana,Arial" COLOR="#ffff33" SIZE="-1">
      Price</FONT></TD>
    <TH><FONT FACE="Verdana,Arial" COLOR="#ffff33" SIZE="-1">
      Extension</FONT></TD>
  </TR>
<%
  cSubTotal = 0
  nNumItems = 0
  for i = 1 to nNumCartItems
    Set rsProducts = Server.CreateObject("ADODB.Recordset")
    sSQL = "SELECT * FROM products WHERE itemproductnumber = '" &
       ↳  sCartItemNums(i) & "'"
    rsProducts.open sSQL, dbMain, adopenstatic,
       ↳  adlockpessimistic, adcmdtext
%>
  <TR BGCOLOR="#99ccff">
<%
  sTemp = "<INPUT TYPE='Text' NAME='txtItemQty" & i & "' VALUE='" &
       ↳  nCartItemQtys(i) & "' SIZE='3'>"
  nNumItems = nNumItems + nCartItemQtys(i)
%>
    <TD ALIGN="center">
      <FONT FACE="Verdana,Arial" COLOR="#000000" SIZE="-1">
      <% = sTemp %></FONT></TD>
```

```
    <TD ALIGN="center">
      <FONT FACE="Verdana,Arial" COLOR="#000000" SIZE="-1">
      <% = rsProducts("itemproductnumber") %></FONT></TD>
    <TD><FONT FACE="Verdana,Arial" COLOR="#000000" SIZE="-1">
      <% = rsProducts("itemname") %></FONT></TD>
<%
  if rsProducts("ItemSalePrice") > 0 then
    cPrice = rsProducts("ItemSalePrice")
  else
    cPrice = rsProducts("ItemRegularPrice")
  end if
%>
    <TD ALIGN="center">
      <FONT FACE="Verdana,Arial" COLOR="#000000" SIZE="-1">
      <% = formatcurrency(cPrice) %></FONT></TD>
<%
  cExtension = nCartItemQtys(i) * cPrice
  cSubTotal = cSubTotal + cExtension
%>
    <TD ALIGN="right">
      <FONT FACE="Verdana,Arial" COLOR="#000000" SIZE="-1">
      <% = formatcurrency(cExtension) %></FONT></TD>
  </TR>
<% next %>
  <TR BGCOLOR="#c5c5c5">
    <TD ALIGN="center" COLSPAN="4">
      <FONT FACE="Verdana,Arial" SIZE="-1">Subtotal</FONT></TD>
    <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
      <% = formatcurrency(cSubTotal) %></FONT></TD>
  </TR>
<% cShipping = 4.50 + (1.25 * nNumItems) %>
  <TR BGCOLOR="#c5c5c5">
    <TD ALIGN="center" COLSPAN="4">
      <FONT FACE="Verdana,Arial" SIZE="-1">Shipping & Handling</FONT></TD>
    <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
      <% = formatcurrency(cShipping) %></FONT></TD>
  </TR>
<% cTotal = cSubTotal + cShipping %>
  <TR BGCOLOR="#66ff00">
    <TD ALIGN="center" COLSPAN= "4">
      <FONT FACE="Verdana,Arial" SIZE="-1">Grand Total</FONT></TD>
    <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
      <% = formatcurrency(cTotal) %></FONT></TD>
  </TR>
</TABLE>
<BR><DIV ALIGN="center">
```

```
<INPUT TYPE="Submit" NAME="cmdSubmit" VALUE="Continue Shopping">
<INPUT TYPE="Submit" NAME="cmdSubmit" VALUE="Recalculate Order">
<BR><BR>
<% if sMode = "checkout" then %>
<INPUT TYPE="Submit" NAME="cmdSubmit" VALUE="Submit Order">
<% else %>
<INPUT TYPE="Submit" NAME="cmdSubmit" VALUE="Proceed to Checkout">
<% end if %>
</DIV></FORM>
<% end if %>

<!--#INCLUDE FILE="footer.inc"-->
</BODY>
</HTML>
```

# מבנה כללי של shopcart.asp

הקובץ shopcart.asp בנוי ממספר שגרות ASP המעורבות בקוד HTML רגיל. קרוב לראשית הדף נקבע מצב ההפעלה הנוכחי, על ידי בדיקת אופן טעינת הדף. אם הדף נטען תוך שימוש במצב מחרוזת שאילתה (Querystring), מופעל מצב הוספה (Add). אחרת, יכול להיות שהדף נטען מטופס המשתמש באחת מקבוצות לחצני Submit בשם cmdSubmit. shopcart.asp יכול לקבוע איזה מצב צריך להיות מופעל, על ידי בדיקת הערך של Request.Form("cmdSubmit"). בנקודות שונות בדף מופעלים קטעי קוד אלה ואחרים, בהתאם למצב ההפעלה הנוכחי.

כל המצבים כוללים טופס HTML; ברוב המצבים יהיה יעדו של action בטופס הקובץ shopcart.asp עצמו. הסיבה העיקרית לרקורסיה זו היא שאם המשתמש משנה את הכמות של פריט אחד או יותר בסל הקניות שלו, עומדת בפניך האפשרות להפעיל קוד ב-shopcart.asp, כדי לשנות את משתני ה-session של סל הקניות, לפני ההפניה לדף אחר.

אם בילית מספיק זמן בשיטוט בין חנויות מקוונות, ייתכן ושמת לב שכאשר משתמש צופה בתוכן סל הקניות שלו הוא יכול לשנות את כמות הפריטים שהוא מתכוון לרכוש; אבל, אם הוא עושה זאת, עליו ללחוץ על לחצן מיוחד לשמירת השינויים בכמויות. באופן אישי אני מוצא זאת נוגד את ההיגיון, במיוחד כאשר המשתמשים אינם משתמשי מחשב מנוסים. השיטה אותה יישמת ב-shopcart.asp מבטלת צורך זה.

# מצב Add

בדרך כלל, הפעם הראשונה ש-shopcart.asp נטען היא כאשר המשתמש לחץ על קישור Add to Cart ליד מוצר כלשהו, אותו הוא מעוניין לרכוש. הקישור Add to Cart מעביר שתי מחרוזות שאילתה ל-shopcart.asp, כפי שמוצג בדוגמה הבאה:

```
<A HREF='shopcart.asp?mode=add&itemno=LMS-0218'>
```

כאשר shopcart.asp מופעל, מפעילה מחרוזת השאילתה mode=add את מצב ההוספה. כתוצאה מכך מופעל קטע הקוד המתחיל בשורה "if sMode = "add. קטע קוד זה סוקר את המערך sCartItemNums, אשר מועבר לכל אורך ה-session כחלק מסל הקניות של המשתמש, כדי לראות אם מספר המוצר המועבר באמצעות מחרוזת השאילתה itemno כבר קיים במערך. אם מספר הפריט קיים במערך, מקודם ערך כמות המוצר; אחרת, המערך גדל כדי לכלול גם את המוצר הזה.

לאחר שבוצעה התוספת הזו לסל הקניות, מתבצע דילוג על פני רוב יתרת קוד ASP שבקובץ, מפני שקוד זה מתייחס למצבי הפעלה אחרים. יתרת shopcart.asp מציגה את התוכן הנוכחי של סל הקניות של המשתמש, כפי שהוצג בתרשים 28.3.

## הצגת סל הקניות

קרוב לתחתית הקובץ shopcart.asp נמצא קטע קוד, המציג את תוכן סל הקניות של המשתמש. הדבר נעשה על ידי שימוש בתערובת של קוד ASP עם קוד HTML רגיל, והוא אמור להיות ברור מאליו כאשר מתבוננים בקוד. בעודו עובר על הפריטים הנכללים במערכים המייצגים את סל הקניות, הוא מאתר כל פריט במסד הנתונים, מציג את המידע אודות פריט זה ומחשב את הסך הכללי הנוכחי. לאחר שכל הפריטים בסל הקניות מוצגים, מחושבות גם הוצאות המשלוח (בהתבסס על משתני ה-session, shipbasecharge ו-shipitemcharge שתוארו בסעיף העוסק בקובץ Global.asa קודם לכן בפרק זה), ומוצגות על המסך. לסיום, מוצג סך כל הרכישה.

## מצב Continue

מצב זה נכנס לפעולה כאשר המשתמש לוחץ על לחצן Continue Shopping. לחיצה זו שולחת את הטופס של cartshop.asp חזרה ל-cartshop.asp עצמו, כפי שתואר קודם לכן. מופעל קוד כדי להגיב לשינויים שעשה המשתמש בכמויות הפריטים ולעדכן את משתני ה-session של סל הקניות. אז, מופנה המשתמש חזרה ל-shopcart.asp להמשך הקנייה.

## מצב Recalculate

מצב זה לא ממש דרוש, אבל הוא מספק למשתמש את האפשרות לראות באופן מיידי את התוצאות של כל שינוי כמות אותו ביצע. הדמיון בין מצב Recalculate (חישוב מחדש) למצב Continue הוא בכך שקוד עדכון סל הקניות בשני המצבים זהה. אבל, במקום להפנות את המשתמש חזרה לדף shopcart.asp, מצב Recalculate מאפשר ליתרת הקוד של shopcart.asp להיות מופעל, וכתוצאה מכך צופה המשתמש בשינויים המעודכנים שערך בסל הקניות המחושב מחדש.

## מצב Checkout

כאשר המשתמש לוחץ על הלחצן Proceed to Checkout, נטען shopcart.asp מחדש במצב Checkout (יציאה). זהו מצב דומה למצב Recalculate, עם שני הבדלים עיקריים.

ההבדל הראשון הוא בהגדרת הטופס עצמו. כל יתר המצבים משתמשים ב-shopcart.asp כיעד עבור ה-action של הטופס. מצב Checkout מפנה את הטופס ליעד אחר, לדף submitorder.asp. כפי שתראה בסעיף הבא, **השלמת ההזמנה**, דף זה יסכם את הזמנת הלקוח ויוסיף אותה למסד הנתונים.

ההבדל השני הוא שכאשר shopcart.asp מופעל במצב Checkout מוצגות בפני המשתמש סדרה של תיבות טקסט המאפשרות לו להקליד את נתוני הכתובת למשלוח ואופן התשלום. נתונים אלה ישמשו את submitorder.asp בעת השלמת ההזמנה.

תרשים 28.4 מציג את shopcart.asp במצב Checkout.

**תרשים 28.4** כאשר המשתמש מוכן להגיע לקופה, עליו להקליד את כתובת המשלוח ולהסדיר את אופן התשלום.

# השלמת ההזמנה

לאחר שהמשתמש הקליד את הכתובת למשלוח המוצרים ואת אופן התשלום המבוקש בקובץ shopcart.asp, הוא ילחץ על לחצן Submit Order, כדי להפעיל את submitorder.asp (ראה תדפיס 28.8). דף זה אחראי להוספת נתוני ההזמנה למסד הנתונים ויספק סיכום של המכירה אותו יוכל המשתמש להדפיס.

```
<!--#INCLUDE FILE="adovbs.inc"-->
<%
 dim sErrorString
 sErrorString = ""
 dim cItemPrice(99), sItemName(99)
 dim cOrderTotal
 cItemPrice(0) = 0
 sItemName(0) = ""
 cOrderTotal = 0

 if session("salecomplete") = true then
   session("salecomplete") = false
   response.redirect "default.asp"
 end if

 sCartItemNums = session("cartitemnumarray")
 nCartItemQtys = session("cartitemqtyarray")
 nNumCartItems = ubound(sCartItemNums)

 for i = 1 to nNumCartItems
   nThisQty = int(request.form("txtItemQty" & i))
   nCartItemQtys(i) = nThisQty
 next
 session("cartitemnumarray") = sCartItemNums
 session("cartitemqtyarray") = nCartItemQtys

Sub CheckErrors(sField, sItemName)
 if trim(request.form(sField)) = "" then
   sErrorString = sErrorString & "The " & sItemName &
     ↳ " field cannot be blank.<BR>"
 end if
End Sub

 CheckErrors "txtCustomerName","Customer Name"
 CheckErrors "txtAddress","Customer Address"
 CheckErrors "txtCity","City"
 CheckErrors "txtState","State"
 CheckErrors "txtZIP","ZIP"
 CheckErrors "txtPhone","Telephone"
 CheckErrors "optPaymentType","Payment Type"
 CheckErrors "txtAcctNo","Account Number"
 CheckErrors "txtExpDate","Expiration Date"

 if sErrorString = "" then

   set dbMain = server.createobject("ADODB.Connection")
   dbMain.open session("dblogin")
```

```
Set rsProducts = Server.CreateObject("ADODB.Recordset")
for i = 1 to nNumCartItems
  sSQL = "SELECT * FROM products WHERE itemproductnumber = '" &
     ↳ sCartItemNums(i) & "'"
  rsProducts.open sSQL, dbMain, adopenstatic,
     ↳ adlockpessimistic, adcmdtext
  if rsProducts("itemsaleprice") > 0 then
    cItemPrice(i) = rsProducts("itemsaleprice")
  else
    cItemPrice(i) = rsProducts("itemregularprice")
  end if
  sItemName(i) = rsProducts("itemname")
  cOrderTotal = cOrderTotal + (nCartItemQtys(i) * cItemPrice(i))
  nNumItems = nNumItems + nCartItemQtys(i)
  rsProducts.close
next
cShipping = session("shipbasecharge") +
   ↳ (nNumItems * session("shipitemcharge"))
cOrderTotal = cOrderTotal + cShipping

dbMain.BeginTrans

Set rsOrders = Server.CreateObject("ADODB.Recordset")
sSQL = "SELECT * FROM orders"
rsOrders.open sSQL, dbMain, adOpenStatic, adlockpessimistic, adcmdtext

rsOrders.addnew
rsOrders("orderdate") = date()
rsOrders("ordertime") = time()
rsOrders("customername") = request.form("txtCustomerName")
rsOrders("customeraddress") = request.form("txtAddress")
rsOrders("customercity") = request.form("txtCity")
rsOrders("customerstate") = request.form("txtState")
rsOrders("customerzip") = request.form("txtZIP")
rsOrders("customerphone") = request.form("txtPhone")
rsOrders("orderamount") = cOrderTotal
rsOrders("paymentmethod") = request.form("optPaymentType")
rsOrders("paymentaccount") = request.form("txtAcctNo")
rsOrders("expdate") = request.form("txtExpDate")

rsOrders.update
rsorders.movelast
nOrderID = rsorders("orderid")
rsOrders.close

Set rsOrderLines = Server.CreateObject("ADODB.Recordset")
```

```
sSQL = "SELECT * FROM orderdetails WHERE orderid = " & nOrderID
    rsOrderLines.open sSQL, dbMain, adOpenStatic,
      ➥ adlockpessimistic, adcmdtext

   for i = 1 to nNumCartItems
      rsOrderLines.addnew
      rsOrderLines("orderid") = nOrderID
      rsOrderLines("productnumber") = sCartItemNums(i)
      rsOrderLines("quantity") = nCartItemQtys(i)
      rsOrderLines("price") = cItemPrice(i)
      rsOrderLines.update
    next
    dbMain.CommitTrans

    session("salecomplete") = true
 end if

  sPageHead = "ASP Unleashed Online Store Order Confirmation"
  sPageCaption = "Order Confirmation"

%>
<HTML>

<!--#INCLUDE FILE="header.inc"-->
<FONT FACE="Verdana,Arial" COLOR="#008000" SIZE="-1">
<%
  if sErrorString = "" then
    response.write "Order # " & nOrderID & " has been entered. Use " &
      ➥ " your browser's PRINT function to print a copy of this order." &
      ➥ "<BR><BR>"
    sSQL = "SELECT * FROM orders WHERE orderid = " & nOrderID
    rsOrders.open sSQL, dbMain, adOpenStatic, adlockpessimistic, adcmdtext
%>
<TABLE ALIGN="center" BORDER="0">
  <TR>
    <TD VALIGN="top">
      <TABLE BORDER="0">
        <TR>
          <TD ALIGN="center" BGCOLOR="#808080">
            <FONT COLOR="#ffffff"><B>Order # <% = nOrderID %>
            </B></FONT></TD>
        </TR>
        <TR>
          <TD ALIGN="center" BGCOLOR="#3300ff">
            <FONT COLOR="#ffffff"><B>Order Date <% = date() %>
            </B></FONT></TD>
        </TR>
      </TABLE>
```

```
    </TD>
    <TD>
      <TABLE ALIGN="center" BORDER="0">
        <TR BGCOLOR="#c0c0c0">
          <TH COLSPAN="2" align="center">
            <FONT FACE="Verdana,Arial" SIZE="-1">
            Customer Information</TD>
        </TR>
        <TR>
          <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
            Name:</TD>
          <TD><FONT FACE="Verdana,Arial" SIZE="-1" COLOR="#0000ff">
            <% = rsOrders("customername") %></TD>
        </TR>
        <TR>
          <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
            Address:</TD>
          <TD><FONT FACE="Verdana,Arial" SIZE="-1" COLOR="#0000ff">
            <% = rsOrders("customeraddress") %></TD>
        </TR>
        <TR>
          <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
            City/State/ZIP:</TD>
          <TD><FONT FACE="Verdana,Arial" SIZE="-1" COLOR="#0000ff">
            <% = rsOrders("customercity") & ", " %>
            <% = rsOrders("customerstate") %>
            <% = " " & rsOrders("customerzip") %></TD>
        </TR>
        <TR>
          <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
            Daytime Phone:</TD>
          <TD><FONT FACE="Verdana,Arial" SIZE="-1" COLOR="#0000ff">
            <% = rsOrders("customerphone") %></TD>
        </TR>
        <TR>
          <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
            Payment Method:</TD>
<%
        select case rsOrders("paymentmethod")
          case "visa"
            sTemp = "Visa"
          case "mc"
            sTemp = "MasterCard"
          case "amex"
            sTemp = "American Express"
        end select
%>
```

```
        <TD><FONT FACE="Verdana,Arial" SIZE="-1" COLOR="#0000ff">
          <% = sTemp %></TD>
        </TD>
      </TR>
    </TABLE>
  </TD>
</TABLE>
<BR><BR>

<TABLE ALIGN="center" BORDER="0">
  <TR BGCOLOR="#ff0066">
    <TH><FONT FACE="Verdana,Arial" COLOR="#ffff33" SIZE="-1">
      Quantity</FONT></TD>
    <TH><FONT FACE="Verdana,Arial" COLOR="#ffff33" SIZE="-1">
      Item No.</FONT></TD>
    <TH><FONT FACE="Verdana,Arial" COLOR="#ffff33" SIZE="-1">
      Item Name</FONT></TD>
    <TH><FONT FACE="Verdana,Arial" COLOR="#ffff33" SIZE="-1">
      Price</FONT></TD>
    <TH><FONT FACE="Verdana,Arial" COLOR="#ffff33" SIZE="-1">
      Extension</FONT></TD>
  </TR>
<%
  cSubTotal = 0
  nNumItems = 0
  for i = 1 to nNumCartItems
%>
  <TR BGCOLOR="#99ccff">
    <TD ALIGN="center">
      <FONT FACE="Verdana,Arial" COLOR="#000000" SIZE="-1">
      <% = nCartItemQtys(i) %></FONT></TD>
    <TD ALIGN="center">
      <FONT FACE="Verdana,Arial" COLOR="#000000" SIZE="-1">
      <% = sCartItemNums(i) %></FONT></TD>
    <TD><FONT FACE="Verdana,Arial" COLOR="#000000" SIZE="-1">
      <% = sItemName(i) %></FONT></TD>
    <TD ALIGN="center">
      <FONT FACE="Verdana,Arial" COLOR="#000000" SIZE="-1">
      <% = formatcurrency(cItemPrice(i)) %></FONT></TD>
<%
  nNumItems = nNumItems + nCartItemQtys(i)
  cExtension = nCartItemQtys(i) * cItemPrice(i)
  cSubTotal = cSubTotal + cExtension
%>
    <TD ALIGN="right">
      <FONT FACE="Verdana,Arial" COLOR="#000000" SIZE="-1">
      <% = formatcurrency(cExtension) %></FONT></TD>
  </TR>
```

```
<% next %>
  <TR BGCOLOR="#c5c5c5">
    <TD ALIGN="center" COLSPAN="4">
      <FONT FACE="Verdana,Arial" SIZE="-1">Subtotal</FONT></TD>
    <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
      <% = formatcurrency(cSubTotal) %></FONT></TD>
  </TR>
<%
  cShipping = session("shipbasecharge") +
    ➥ (nNumItems * session("shipitemcharge"))
%>
  <TR BGCOLOR="#c5c5c5">
    <TD ALIGN="center" COLSPAN="4">
      <FONT FACE="Verdana,Arial" SIZE="-1">Shipping & Handling</FONT></TD>
    <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
      <% = formatcurrency(cShipping) %></FONT></TD>
  </TR>
<% cTotal = cSubTotal + cShipping %>
  <TR BGCOLOR="#66ff00">
    <TD ALIGN="center" COLSPAN= "4">
      <FONT FACE="Verdana,Arial" SIZE="-1">Grand Total</FONT></TD>
    <TD ALIGN="right"><FONT FACE="Verdana,Arial" SIZE="-1">
      <% = formatcurrency(cTotal) %></FONT></TD>
  </TR>
</TABLE>

<DIV ALIGN="center"><FONT FACE="Verdana,Arial" SIZE="+1">
<BR>Thank you for your order!</FONT></DIV>
<% else %>
The following errors error(s) occurred while processing the information
you provided:<BR><BR>
<FONT FACE="Courier New" COLOR="#000000">
<% = sErrorString %>
</FONT>
<BR><BR>Please press your browser's BACK button and correct this
information, then resubmit your order.
<% end if %>
<BR>
<%
  if session("salecomplete") = true then
    redim sItemNums(0)
    redim nItemQtys(0)
    sItemNums(0) = ""
    nItemQtys(0) = 0
    session("CartItemNumArray") = sItemNums
    session("CartItemQtyArray") = nItemQtys
    session("salecomplete") = false
  end if
%>
<!--#INCLUDE FILE="footer.inc"-->
</BODY>
</HTML>
```

## התחלת התהליך

לאחר הגדרת חלק מהמשתנים, מכיל submitorder.asp פיסת קוד מעניינת:

```
if session("salecomplete") = true then
    session("salecomplete") = false
    response.redirect "default.asp"
end if
```

קוד זה הוא אמצעי זהירות שנועד למנוע מהמשתמש לשלוח בטעות שוב את אותו טופס הזמנה, במידה וטען מחדש בשוגג את submitorder.asp, מסיבה כלשהי. כפי שתוכל לראות בהמשך הקוד, ל-salecomplete, משתנה ה-session, מוגדר הערך True לאחר שההזמנה הושלמה ואוחסנה במסד הנתונים. הקוד שלאחר מכן בוחן את הערך של session("salecomplete"); אם הוא True, ההזמנה הושלמה ואין צורך להמשיך ולהפעיל את הקוד בדף זה. במקרה זה יופנה המשתמש לדף default.asp.

לאחר שעניין זה טופל, מפעיל submitorder.asp את קוד החישוב מחדש של סל הקניות, שבקובץ SHOPCART.ASP, למקרה שהמשתמש שינה כמות פריטים כלשהי בתיבות הטקסט שבקובץ shopcart.asp, לפני שלחץ על לחצן Submit Order.

## בדיקת שגיאות

כעת, מוודא submitorder.asp שבכל תיבת טקסט בחלקו העליון של shopcart.asp הוקלדה מחרוזת כלשהי. כאן צצה לפניך הזדמנות פז להתאים את יישום החנות המקוונת בהתאם לצרכי הארגון שלך. למשל, תוכל לאפשר ללקוח שלך לבצע כניסה למערכת (Login) או לאחזר את נתוני המשתמש מבסיס הלקוחות של הארגון. בנוסף, בהתאם לאופן בו מוגדרת מערכת ההזמנות של הארגון שלך, תוכל לוודא באופן כלשהו את אמיתות פרטי כרטיס האשראי של הלקוח, לפני שתמשיך.

בדוגמה שלפנינו, אם המשתמש השאיר תיבת טקסט כלשהי ריקה, מופיעה הודעה המוסרת לו איזו מהן גרמה לבעיה ומורה לו לחזור לדף הקודם ולהשלים את המידע החסר.

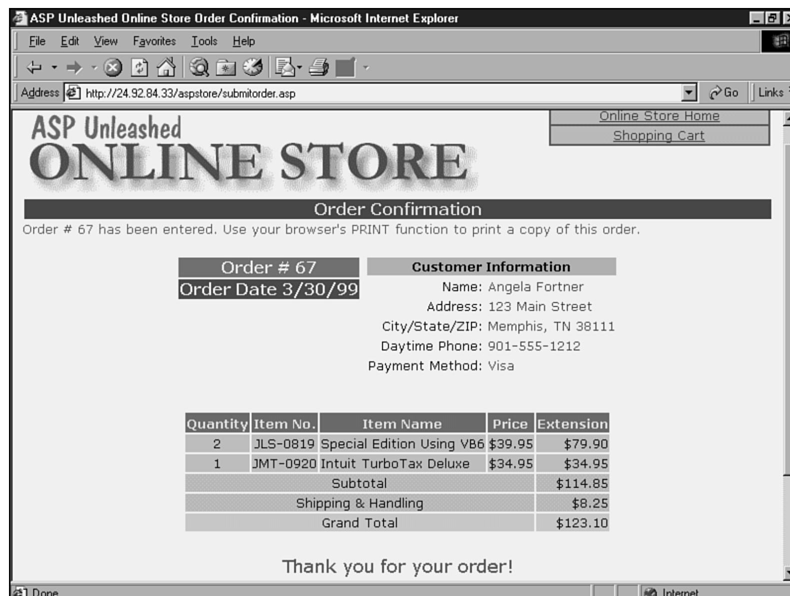
## הוספת ההזמנה

לאחר שהנתונים שסופקו על ידי המשתמש אומתו, אתה מוכן ומזומן לסכם את המכירה, על ידי הוספת רשומה אחת עבור ההזמנה כולה לטבלה Orders, ורשומה אחת עבור כל פריט שהוזמן לטבלה Order Details. דבר זה מושג על ידי מעבר על משתני ה-session של סל הקניות ושימוש בטכניקות ADO סטנדרטיות להוספת הרשומות המתאימות.

יכול להיות שתצטרך לפתח סוג כלשהו של ממשק למסד הנתונים ממנו תוכל לחלץ נתוני הזמנה כאשר ההזמנות נרשמות. לאחר שנודע לך כי בוצעו הזמנות חדשות תוכל להתחיל בתהליך הטיפול בהזמנות.

## הצגת הזמנה

לאחר שנתוני ההזמנה נרשמו מוצג על המסך סיכום של ההזמנה (כולל מזהה הזמנה שחושב זה עתה, ותאריך ביצוע ההזמנה), והמשתמש מוזמן להדפיס אותו. הדבר מוצג באופן דומה לזה בו מוצג סל הקניות ב-shopcart.asp. תרשים 28.5 מציג את ההזמנה המושלמת כפי שמציג אותה submitorder.asp.




**תרשים 28.5** המשתמש סיים את הזמנתו וכעת הוא יכול להדפיס עותק של הרשומות שלו

## ניקיון

לסיום, פיסת הקוד האחרונה ב-submitorder.asp מאתחלת את משתני ה-session של סל הקניות. הדבר נעשה כדי לרוקן את סל הקניות, למקרה והמשתמש מחליט לערוך קניות נוספות. הוא אינו מעוניין לרכוש את אותו פריט פעם נוספת!

# הזדמנויות לשיפורים

כפי שנידון קודם לכן, היישום לדוגמה שהוצג כאן נועד להיות כללי. כאשר תפתח את האתר שלך עליך לנצל את אותם אזורים בהם תוכל לשפר ולהתאים את האתר לצרכיך. חלק מהנושאים העיקריים שיכולים להיות מותאמים כוללים:

- סקירת האפשרויות לרישום יומן למערכת. הדבר יחסוך למשתמש את הצורך להקליד את המידע בכל פעם שהוא מתחבר למערכת שלך.
- הוספת הפונקציה Order Status. כאשר אתה יוצר ממשק עם היישום לתוך מערכת המשלוחים שלך, ייתכן ותוכל לעדכן את מצב ההזמנה בטבלה Orders. במקרה זה, אמורה להיות למשתמש האפשרות לחפש אחר הזמנה שביצע, כדי לראות אם וכיצד היא נשלחה.
- הוספת שגרה המוודאת את נתוני כרטיס האשראי של המשתמש.
- הודעה אוטומטית לגבי קיומן של הזמנות חדשות הנשלחת ישירות למחלקה האחראית על אריזה ומשלוח.

# סיכום

בפרק זה ראית כיצד לפתח חנות מקוונת בסיסית. ראית כיצד להשתמש בקוד ASP יחד עם קוד HTML רגיל, ולמדת מספר עקרונות לזרימה חלקה של האתר, מה שיהפוך את הגלישה באתר ליעילה וקלה עבור המשתמש. בנוסף, הוצגו בפניך מספר רעיונות להתאמה אישית של האתר, כדי שיפעל בהתאם לצרכי הארגון שלך.

## פרק 29

# יצירת אתר לחיפוש עבודה

**בפרק זה:**

- **סקירת אתר לחיפוש עבודה**
- **יצירת האתר**
- **הרחבת הפרויקט**

פרק זה יציג בפניך תכנות יישום ASP מהעולם האמיתי. תלמד כיצד ליצור אתר ASP לחיפוש עבודה, אתר מקוון לחיפוש אחר תעסוקה עבור יועצים ועובדים ב-ASP. יישום תכנות (Programming Application) זה מאפשר ליועצי ASP לפרסם את קורות החיים שלהם ולתור אחר אפשרויות תעסוקה. מעסיקים יכולים להיעזר באתר זה כדי לפרסם רשימת דרושים למשרות פנויות, ולאתר עובדים באמצעות קורות חיים המתפרסמים באתר.

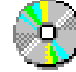
אתר למחפשי עבודה של ASP מציג כיצד ליצור אתר אינטרנט מבוסס מסד נתונים תוך שימוש בדפי ASP. כל המידע, הן של קורות החיים והן של המשרות הפנויות, מאוחסן כולו במסד נתונים של שרת SQL. יישום זה גם מדגים כיצד לעבוד עם נתוני טופס. תלמד כיצד להשתמש בדפי ASP כדי לאסוף ולהציג מידע שהוקלד לטפסי HTML.

# סקירת אתר לחיפוש עבודה

אתר חיפוש העבודה של ASP כולל 10 דפי ASP. כל אחד מדפים אלה יידון בסעיפים מתקדמים יותר של הפרק, אחד אחרי השני. אבל, לפני שניכנס לפרטים לגבי אופן פעולתו של כל דף, להלן סקירה מהירה המתארת את הביצוע של כל דף:

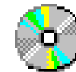
- default.asp - זהו דף הבית של ASP Job Site. יש בו סקירה אודות תוכן האתר ותפריט לדפים האחרים.
- postresume.asp - באמצעות דף זה יכולים המבקרים באתר לפרסם את קורות החיים שלהם על ידי מילוי טופס HTML.
- postresume2.asp - דף זה מכניס רשומת קורות חיים חדשה למסד הנתונים ומודה למשתמש על פרסום קורות חייהם.
- postjob.asp - באמצעות דף זה יכולים מעסיקים המבקרים באתר לפרסם את המשרות הפנויות בארגון שלהם, על ידי מילוי טופס HTML.
- postjob2.asp - דף זה רושם רשומת משרה פנויה חדשה למסד הנתונים ומודה למשתמש על פרסום הצעת העבודה.
- searchresumes.asp - דף זה מאפשר למעסיקים לבחון את כל רשומות קורות החיים המפורסמות באתר.
- resumedetail.asp - דף זה מציג את קורות החיים המלאים של המועמד, על ידי אחזורם ממסד הנתונים.
- searchjobs.asp - דף זה מאפשר ליועצי ASP לחפש בין כל הצעות העבודה באתר.
- jobdetail.asp - דף זה מציג את הפרטים המלאים אודות הצעת העבודה הנבחרת, על ידי אחזורם ממסד הנתונים.
- jobsitefuncs.asp - דף זה מכיל את כל השגרות והפונקציות השכיחות בהן משתמשים דפי ASP אחרים באתר חיפוש העבודה.

# התקנת האתר



לפני שתמשיך לקרוא בפרק, יהיה עליך להתקין את האתר ASP Job Site בשרת האינטרנט שלך. כדי להתקין את האתר העתק את כל הקבצים מהתיקיה Chapter29 שבתקליטור המצורף לספר, לתיקיה בשרת האינטרנט שלך. יש לאפשר לשרת האינטרנט גישה לתיקיה זו. למשל, תוכל להעתיק את הקבצים לתיקיה בשם JobSite הממוקמת מתחת לתיקיית השורש של שרת האינטרנט שלך (בדרך כלל זו התיקיה wwwroot).

בנוסף, עליך ליצור הרשאות Script ו-Execute לתיקיה בה מאוחסנים קבצי האתר. כדי לעשות זאת עליך להפעיל את Internet Service Manager ולפתוח את תיבת הדו-שיח Properties של תיקיית האתר. בחר בכרטיסיה Directory וסמן את תיבות הסימון Script ו-Execute.



לאחר שהעתקת את קבצי האתר לשרת האינטרנט עליך ליצור שתי טבלאות מסד נתונים בשמות Resume ו-Jobs. בטבלאות אלו יישמרו הנתונים של רשומות קורות החיים והצעות העבודה. בתיקיה Chapter29 שבתקליטור תמצא קובץ בשם maketables.sql אשר ייצור טבלאות אלו עבורך באופן אוטומטי. טען את הקובץ maketables.sql לתוך Microsoft Query Analyser (או SQL/w אם אתה משתמש ב-SQL גירסה 6.5) והפעל את הקובץ.

הקובץ maketables.sql יוצר את הטבלאות על ידי הפעלת הצהרת SQL הנראית בתדפיס 29.1.

**תדפיס 29.1** maketables.sql

```
CREATE TABLE [dbo].[Jobs] (
    [job_id] [int] IDENTITY (1, 1) NOT NULL ,
    [fname] [varchar] (100) NULL ,
    [lname] [varchar] (100) NULL ,
    [phone] [varchar] (20) NOT NULL ,
    [email] [varchar] (80) NOT NULL ,
    [state] [varchar] (5) NULL ,
    [country] [varchar] (5) NULL ,
    [bdesc] [varchar] (100) NULL ,
    [perm] [varchar] (3) NULL ,
    [contract] [varchar] (3) NULL ,
    [job] [text] NULL ,
    [expires] [datetime] NULL ,
    [entrydate] [datetime] NOT NULL DEFAULT getDATE()
)
GO
```

```
grant all on Jobs to public

/****** Object:  Table [dbo].[Resumes]    Script Date: 3/21/99 3:05:35 AM ******/
CREATE TABLE [dbo].[Resumes] (
    [resume_id] [int] IDENTITY (1, 1) NOT NULL ,
    [fname] [varchar] (100) NULL ,
    [lname] [varchar] (100) NULL ,
    [phone] [varchar] (20) NULL ,
    [email] [varchar] (80) NULL ,
    [state] [varchar] (5) NULL ,
    [country] [varchar] (5) NULL ,
    [bdesc] [varchar] (100) NULL ,
    [perm] [varchar] (3) NULL ,
    [contract] [varchar] (3) NULL ,
    [resume] [text] NULL ,
    [expires] [datetime] NULL ,
    [entrydate] [datetime] NOT NULL DEFAULT getDATE()
)
GO
```

לאחר שהוספת את הטבלאות תוכל להגדיר את האתר ASP Job Site לגשת אליהן. כל הקבצים משתמשים בקבוע גלובלי בשם connectionstring כדי לפתוח את החיבור (connection) למסד הנתונים. קבוע (Constant) זה מוגדר בשורה הראשונה של הקובץ jobsitefuncs.asp. השורה נראית כך:

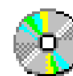
```
CONST connectionstring =
    ↳ "PROVIDER=SQLOLEDB;UID=sa;PWD=secret;DATABASE=test"
```

החלף את ערכי הפרמטרים של UID ושל PWD בערכי אבטחה חוקיים למסד הנתונים שלך. אחר כך, שנה את השם DATABASE לשם מסד נתונים הקיים בשרת שלך (זה אמור להיות אותו בסיס נתונים בו יצרת את הטבלאות Jobs ו-Resume).

**הערה:**

יש יתרון בשימוש במחרוזת חיבור (string connection) מכיון שאז איננו צריכים ליצור DSN בשרת (למידע נוסף פנה לפרק 12).

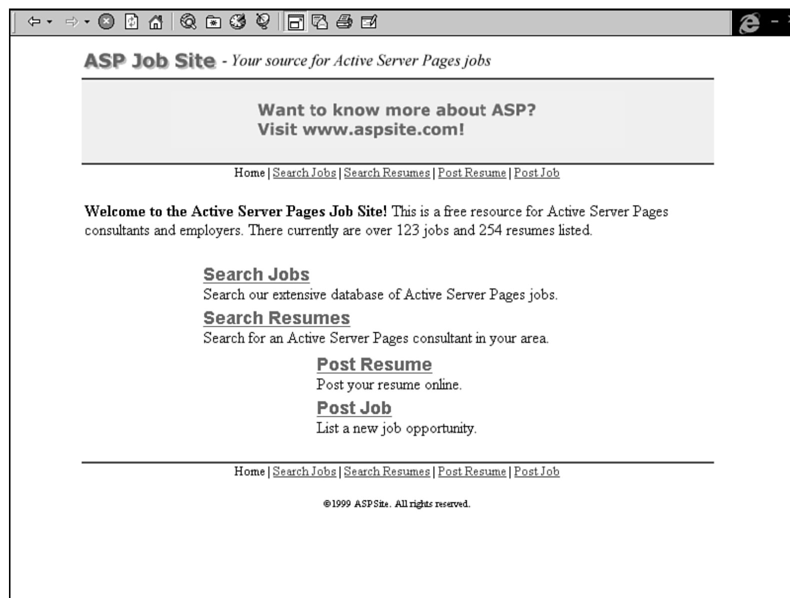
# יצירת האתר



הסעיפים הבאים מפרטים בהרחבה כיצד פועל כל אחד מהקבצים באתר ASP Job Site. מכיון שקבצים רבים ארוכים מכדי שניתן יהיה להציג את התדפיס שלהם בספר באופן מלא, אני ממליץ לקורא לפתוח עורך טקסט כלשהו (כגון פנקס הרשימות, Notepad) ולפתוח את הקבצים בעודך קורא פרק זה. כל הקבצים קיימים כאמור בתיקיה Chapter29 שבתקליטור המלווה את הספר.

# הצגת דף הפתיחה

דף הפתיחה של ASP Job Site נמצא בדף default.asp. מטרת דף הפתיחה היא לברך מבקרים חדשים באתר ולספק תפריט קישורים לדפים האחרים באתר (ראה תרשים 29.1). בנוסף הוא גם מציג סטטיסטיקות לגבי מספר רשומות קורות החיים והצעות העבודה שבאתר.




**תרשים 29.1** דף הפתיחה של ASP Job Site

תוכן HTML הנכלל בחלקו העליון ובחלקו התחתון של דף זה נוצר על ידי שתי שגרות, ששמן displayHeader ו-displayFooter. אותן שתי שגרות משמשות ליצירת הכותרות העליונה והתחתונה בכל דפי האתר, ותוכל למצוא אותן בקובץ jobsitefuns.asp, אשר נכלל בכל דף באמצעות הפניית INCLUDE של ASP.

הכותרת העליונה כוללת פרסומת. פרסומת זו נוצרת על ידי השימוש הרכיב Ad Rotator. רכיב זה מאפשר לך להציג באופן אקראי כרזות פרסום שונות, במרווחי זמן שונים. כדי ללמוד ביתר הרחבה אודות הרכיב Ad Rotator פנה לפרק 9.

גם הכותרת העליונה וגם התחתונה כוללות סרגל ניווט המציג קישורים לדפי ASP אחרים באתר. שמו של כל דף, חוץ מאשר של הדף הנוכחי, מופיע כקישור טקסט (Text Hyperlink). סרגל הניווט מאפשר למשתמשים גישה קלה לכל אחד מהדפים, וגם מציג מידע כללי אודות הדף הנוכחי המופיע בפני המשתמש. סרגל הניווט נוצר על ידי השגרה שבתדפיס 29.2 (שים לב שהסימן | המופיע בקוד להלן הוא מקש ¦ במקלדת).

```
Sub displayNavbar( byVal currentPage )
    if currentPage = "Home" then
    %>
    Home |
    <%
    else
    %>
    <A HREF="default.asp">Home</A> |
    <%
    end if
    if currentPage = "Search Jobs" then
    %>
    Search Jobs |
    <%
    else
    %>
    <A HREF="searchjobs.asp">Search Jobs</A> |
    <%
    end if
    if currentPage = "Search Resumes" then
    %>
    Search Resumes |
    <%
    else
    %>
    <A HREF="searchresumes.asp">Search Resumes</A> |
    <%
    end if
    if currentPage = "Post Resume" then
    %>
    Post Resume |
    <%
    else
    %>
    <A HREF="postresume.asp">Post Resume</A> |
    <%
    end if
    if currentPage = "Post Job" then
    %>
    Post Job
    <%
    else
    %>
    <A HREF="postjob.asp">Post Job</A>
    <%
    end if
End Sub
```

תדפיס 29.2 כולל מספר רב של התניות. כאשר נקראת שגרה מועבר אליה שמו של הדף הנוכחי. אם שמו של דף אינו שווה לשמו של הדף הנוכחי, מוצג קישור טקסט לאותו דף. אחרת, מוצג שם הדף בטקסט פשוט (שאינו קישור). דף הפתיחה של האתר מציג נתונים סטטיסטיים לגבי מספר קורות החיים והצעות העבודה המפורסמים באתר. סטטיסטיקה זו נוצרת על ידי התסריט המוצג בתדפיס 29.3.

**תדפיס 29.3** הצגת סטטיסטיקות עבור Job ו-Resume

```
' Get resume and job statistics
if Application( "statupdate" ) = "" OR Application( "statupdate" ) < NOW() then
    Set Con = Server.CreateObject( "ADODB.Connection" )
    Con.Open connectionstring
    mySQL = "SELECT COUNT(*) numresumes FROM Resumes"
    SET RS = Con.Execute( mySQL )
    Application( "resumes" ) = RS( "numresumes" )
    mySQL = "SELECT COUNT(*) numjobs FROM Jobs"
    SET RS = Con.Execute( mySQL )
    Application( "jobs" ) = RS( "numjobs" )
    Application( "statupdate" ) = DATEADD( "h", 1, NOW() )
end if
```

התסריט בתדפיס 29.3 דורש מעט הרחבה. הנתונים הסטטיסטיים לגבי מספר הצעות העבודה וקורות החיים המפורסמים באתר מאוחזרים ממסד הנתונים. הדבר נעשה על ידי פתיחת חיבור מסד נתונים (Database Connection) והפעלת שתי שאילתות (*)COUNT של SQL. פעולה זו יקרה, מפני שפעולה על מסד הנתונים עצמו היא אחת מהפעולות האיטיות ביותר שתוכל לבצע בדף ASP.

כדי למנוע שתי שאילתות למסד הנתונים בכל פעם שמישהו מבקש את דף הפתיחה, מאוחסנים הנתונים הסטטיסטיים בשני משתני יישום (Application Variables), אשר מתעדכנים רק פעם בשעה. מכיון שההנחה היא שנתוני הסטטיסטיקה לא ישתנו לעיתים קרובות, אתה מונע בדרך זו עומס מיותר על שרת מסד הנתונים שלך.

בכל פעם שנתוני הסטטיסטיקה מעודכנים מוגדר ערכו של משתנה בשם statupdate לשעה אחת קדימה. כל עוד השעה הנוכחית היא פחות מהשעה המאוחסנת במשתנה זה, התסריט לא יופעל. בשל כך מעודכנים נתוני הסטטיסטיקה רק פעם בשעה.

# רישום קורות חיים

שני דפי ASP, postresume.asp ו-postresume2.asp, מאפשרים למשתמש לרשום את קורות חייו ולשמור אותם במסד הנתונים. הדף postresume.asp מציג טופס HTML לכתיבת קורות חיים. כאשר המשתמש שולח את הטופס, הדף postresume2.asp מוסיף בפועל את הנתונים שבטופס לטבלת מסד נתונים בשם Resumes.

טופס HTML שנוצר על ידי postresume.asp נראה בתרשים 29.2. הוא מאפשר למשתמש להקליד את נתוניו האישיים ואת האמצעים ליצירת קשר עימו, כגון שמו, כתובת הדואר האלקטרוני שלו ומספרי הטלפון שלו. הוא גם מאפשר למשתמשים לבחור את המדינה והארץ מתוך שתי רשימות בחירה של HTML. רשימות המדינות והארצות נוספות לטופס על ידי הכללת שני קבצים ששמותיהם states.inc ו-countries.inc. קבצים אלה מכילים את רשימת המדינות והארצות.

**תרשים 29.2** טופס לכתיבת קורות חיים

הטופס גם מאפשר למשתמשים בו לציין את משך הזמן בו קורות החיים שלהם יפורסמו באתר. המשתמש יכול לבחור להציג את קורות חייו לתקופה שבין שבוע ימים לשמונה שבועות. לאחר שפרק זמן זה חולף, קורות החיים של משתמש זה כבר לא יופיעו בתוצאות החיפוש באתר.

לסיום, הטופס כולל אזור טקסט גדול בו יכול המשתמש להקליד את הטקסט העיקרי של קורות חייו. טקסט זה יכול להיות טקסט פשוט, או להיות מעוצב באמצעות תגיות HTML.

כאשר המשתמש שולח את הטופס נשלחים הנתונים שלו לקובץ postresume2.asp. דף זה מבצע שלוש פעולות. ראשית, הוא בודק אם כל השדות הנדרשים מולאו כהלכה. הדבר נעשה באמצעות התסריט בתדפיס 29.4.

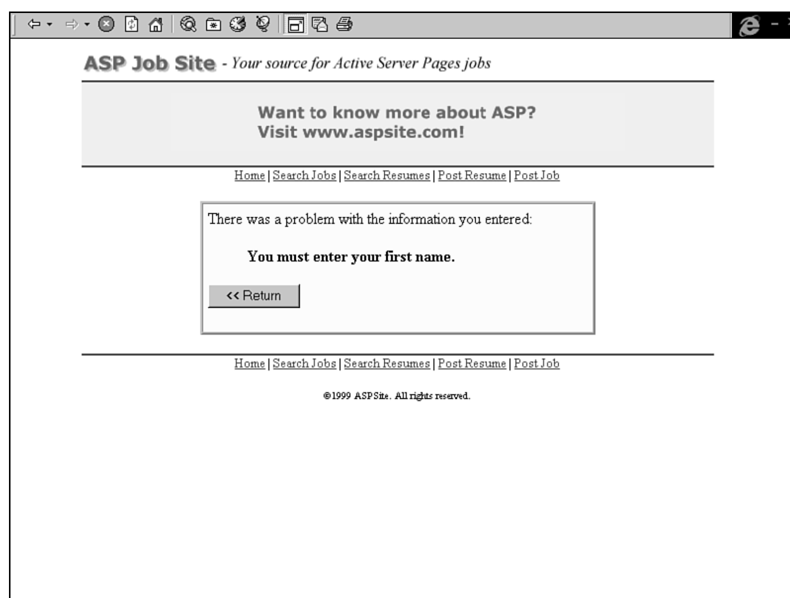
**תדפיס 29.4** בדיקת השדות ההכרחיים בטופס קורות החיים

```
' Check For Required Fields
if TRIM( Request( "fname" ) ) = "" then
    errorForm "postresume.asp", "You must enter your first name."
end if
```

```
if TRIM( Request( "lname" ) ) = "" then
    errorForm "postresume.asp", "You must enter your last name."
end if
if TRIM( Request( "bdesc" ) ) = "" then
    errorForm "postresume.asp", "You must enter a brief description."
end if
if TRIM( Request( "resume" ) ) = "" then
    errorForm "postresume.asp", "You must enter a resume."
end if
```

התסריט בתדפיס 29.4 מאחזר את הערכים של כל שדה טופס נדרש ובודק אם כל אחד מהם שווה למחרוזת באורך 0 (אפס). אם ערך שדה נדרש כלשהו שווה למחרוזת באורך 0 (אפס) נקראת השגרה errorForm.

את השגרה errorForm תמצא בקובץ jobsitefuncs.asp. שגרה זו מציגה הודעת שגיאה ומעבירה את כל נתוני שדות הטופס למשתני שדות נסתרים (Hidden Form Variables) (ראה תרשים 29.3). כאשר המשתמש לוחץ על לחצן Return, מועברים כל הנתונים המקוריים שהוא הקליד בטופס קורות החיים חזרה לטופס ומוצגים בו מחדש.




**תרשים 29.3** טופס שגיאה המופיע כאשר נשלח טופס קורות חיים

כעת, אם משתמש הקליד נתונים בכל השדות הנדרשים, ינסה הקובץ postresume2.asp להוסיף את נתוני הטופס לטבלת מסד הנתונים Resume, באמצעות התסריט שבתדפיס 29.5.

```
' Ready Database Connection
Dim Con
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open ConnectionString

' Check Whether Resume was already submitted
mySQL = "SELECT * FROM Resumes " &
  ↳ "WHERE lname='" & fixQuotes( Request( "lname" ) ) & "' " &
  ↳ "AND bdesc='" & fixQuotes( Request( "bdesc" ) ) & "' " &
  ↳ "AND expires > getDate()"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.Open mySQL, Con, adOpenStatic, adLockOptimistic
if not RS.EOF then
    errorForm "default.asp", "You have already submitted your resume."
end if

' Enter Resume Into Database
RS.AddNew
for each field in Request.Form
    RS( field ) = Request.Form( field )
next
RS.Update
RS.Close
Set RS = Nothing
Set Con = Nothing
```

שים לב שהתסריט בתדפיס 29.5 בעצם מבצע שתי משימות. ראשית, הוא בודק אם אותם קורות חיים כבר מפורסמים באתר. למשתמשי אינטרנט רבים יש הרגל מגונה ללחוץ על לחצן רענן (Refresh או Reload) בדפדפן שלהם לאחר שהם שלחו טופס. אם לא תפעל בנידון, תהיה התוצאה כפילות נתונים במסד הנתונים שלך (מכיון שאותו דף ASP מופעל פעמיים).

כדי לעקוף בעיה כגון זו, בודק התסריט בתדפיס 29.5 האם כבר קיימת במסד הנתונים רשומת קורות חיים של אדם בשם משפחה (Last Name) כזה ועם תיאור (Description) דומה. אם נמצא כי אותם קורות חיים כבר הוגשו לאתר, נקראת השגרה errorForm ומוצג טופס שגיאה.

התסריט בתדפיס 29.5 נעזר בטריק קטן כדי להוסיף את נתוני הטופס למסד הנתונים. שמות העמודות שבטבלה Resume זהים לחלוטין לשמות השדות בטופס HTML. תוך ניצול עובדה זו, הנתונים מועברים למסד הנתונים על ידי ביצוע מעבר עקבי על האוסף Form. טריק זה מאפשר לך להוסיף בקלות מידע לטבלת מסד נתונים מטופס HTML, ללא התחשבות במספר השדות שמכיל הטופס.

# הוספת הצעת עבודה

הוספת הצעת עבודה לאתר מבוצעת על ידי שני דפי ASP, postjob.asp ו-postjob2.asp. שני דפים אלה כמעט וזהים לדפים postresume.asp ו-postresume2.asp שנידונו בסעיף הקודם. אבל, ההבדל ביניהם הוא שנתוני הטפסים הללו מאוחסנים בטבלת מסד הנתונים Jobs, ולא בטבלה Resume.

הדף postjob.asp יוצר את הטופס המוצג בתרשים 29.4. הוא מאפשר למעסיק להקליד דרכי התקשרות ואת הטקסט המתאר את המשרה המוצעת. גם במקרה זה נעשה שימוש בקבצים states.inc ו-countries.inc כדי ליצור את רשימות הבחירה בטופס.

**תרשים 29.4** טופס להוספת הצעת עבודה

הטופס גם מאפשר למשתמשים בו לציין את משך הזמן בו אמורה הצעת העבודה להופיע. המשתמש יכול לבחור להציג את הצעת העבודה למשך שבוע אחד ועד לשמונה שבועות. לאחר פרק זמן זה תפסיק הצעת העבודה להיות מוצגת בתוצאות החיפוש.

כאשר נשלח הטופס postjob.asp נשלחים נתוני הטופס לדף postjob2.asp, וזה האחרון מבצע בו שלוש פעולות. ראשית, הוא בודק אם כל השדות הנדרשים בטופס מולאו כהלכה. לאחר מכן הוא בודק אם כבר קיימת רשומה עבור אותה משרה ולבסוף הוא מוסיף את הנתונים לטבלת מסד הנתונים Jobs.

# חיפוש ברשימת קורות החיים

הדף Search Resumes (הקובץ searchresumes.asp) מאפשר למעסיקים לבצע חיפוש בכל רשומות קורות החיים במסד הנתונים. המשתמש יכול לחפש את קורות החיים על פי מדינה או ארץ, על פי סוג המשרה (מלאה או חלקית) או על ידי הקלדת מילות מפתח לחיפוש (ראה תרשים 29.5).

**תרשים 29.5** דף לחיפוש קורות חיים

כאשר המשתמש שולח את הטופס Search Resumes נשלחים הנתונים לאותו דף עצמו (שיטת הטופס Form Method, היא post והפעולה, Action, היא לאותו דף). הדף Search Resumes מזהה שהטופס נשלח, על ידי כך שהוא בודק את קיומו של שדה טופס נסתר בשם doSearch בטופס Search Resumes. אם לשדה נסתר זה יש ערך, אז מבוצעת שאילתה למסד הנתונים. שאילתת מסד הנתונים בנויה מהתסריט שבתדפיס 29.6.

**תדפיס 29.6** בניית השאילתה עבור Search Resumes

```
' Get Search Criteria
state = TRIM( Request( "state" ) )
country = TRIM( Request( "country" ) )
perman = TRIM( Request( "perman" ) )
contract = TRIM( Request( "contract" ) )
keywords = TRIM( Request( "keywords" ) )

' Create Search String
mySQL = "SELECT * FROM Resumes WHERE expires>getDATE() "
if state <> "??" then
    mySQL = mySQL & "AND state='" & state & "' "
end if
```

```
if country <> "???" then
    mySQL = mySQL & "AND country='" & country & "' "
end if
if perman <> "" AND contract = "" then
    mySQL = mySQL & "AND perman = 'Yes' "
end if
if contract <> "" AND perman = "" then
    mySQL = mySQL & "AND contract = 'Yes' "
end if
if keywords <> "" then
    mySQL = mySQL & "AND bdesc LIKE '%" & fixQuotes( keywords ) & "%' "
end if
mySQL = mySQL & "ORDER BY entrydate DESC"
```

התסריט בתדפיס זה מאחזר את הערכים משדות הטופס שנשלחו מהטופס Search Resumes, משייך ערכים אלה למשתנים מקומיים (Local Variables) ולאחר מכן הוא בונה מחרוזת שאילתת SQL חוקית. למשל, אם המשתמש בוחר להגביל את החיפוש אחר קורות חיים על פי מדינה מסוימת אז המשפט AND state='state name' נוסף לשאילתת SQL שנוצרת.

השאילתה גם מונעת הצגתם של קורות חיים שאינם עדכניים, וזאת באמצעות המשפט WHERE expires>getDATE(). העמודה Expires מכילה את תאריך סיום הצגת קורות החיים. מאחר והפונקציה getDATE() של SQL מחזירה את התאריך והשעה הנוכחיים, משפט זה ימנע הצגת קורות חיים שתוקפם פג לפני התאריך והשעה הנוכחיים.

**הערה:**

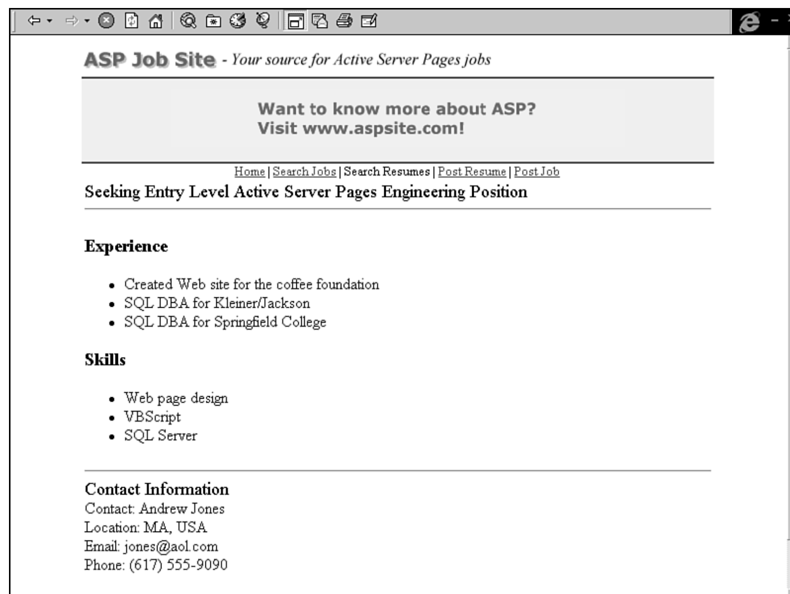
קורות חיים ממשיכים לתפוס נפח דיסק גם לאחר שתוקפם פג. תוכל לבצע מחיקה ידנית של אותם קורות חיים שתוקפם פג מדי תקופת זמן שתקבע לעצמך, או שתוכל ליצור משימת שרת SQL (SQL Server Job) שתבצע זאת באופן אוטומטי.

שים לב שהשאילתה עושה שימוש באופרטור LIKE של SQL כדי למצוא התאמות למילות המפתח שהקליד המשתמש בחיפוש. כפי שלמדת בפרק 15, האופרטור LIKE אינו השיטה היעילה ביותר לביצוע חיפוש. אם אתה משתמש בגירסה 7.0 של שרת SQL של Microsoft, עליך להשתמש בשאילתת טקסט מלא (Full Text Query), ולא באופרטור LIKE.

לאחר ששאילתת SQL נבנית, היא מופעלת ומוצגות תוצאותיה. אותם קורות חיים התואמים לשאילתה מוצגים כעת כרשימה הממוינת על פי התאריך בו הם נשלחו למסד הנתונים. המיון מתבצע על פי תוכן העמודה entrydate שבטבלה Resume. במקום להציג את תוכנם המלא של כל קורות החיים, מופיע רק התיאור המקוצר שכתב מחפש העבודה כקישור. כל קישור שכזה מצביע על דף ASP הנקרא Resume Detail, המציג את כל קורות החיים.

# הצגת קורות חיים

הדף Resume Detail (הקובץ resumedetail.asp) מציג את נתוני קורות החיים, כולל כל דרכי ההתקשרות והטקסט המלא של קורות החיים (ראה תרשים 29.6).




**תרשים 29.6** הצגת פרטי קורות החיים

הדרך היחידה להגיע לדף זה היא על ידי ביצוע חיפוש אחר קורות חיים ולחיצה על אחד מהקישורים המופיעים בתוצאות החיפוש. לכל קישור יש משתנה URL בשם id אשר מזהה באופן ייחודי את קורות החיים. המשתנה id נלקח מהעמודה IDEENTITY resume_id שבטבלה Resumes.

הדף Resume Detail יודע איזה קורות חיים להציג על ידי שימוש במשתנה id. כאשר מתבקש הדף Resume Detail הוא מאחזר את ערך המשתנה id מהאוסף QueryString. אחר כך, הוא מאחזר את קורות החיים המתאימים מהטבלה Resumes, על ידי התאמת המשתנה id עם תוכן העמודה resume_id.

הטקסט של קורות החיים יכול להכיל קוד HTML, ומצד שני הוא יכול גם להכיל טקסט פשוט. הדבר יוצר בעיה בעיצוב תוכן קורות החיים. קוד HTML מתעלם מתווי רווח לבנים (Whitespaces), כגון הקשות Enter ורווחים כפולים בין מילים. אם תציג את תוכן קורות החיים כפי שהם, יאבדו כל הרווחים שהוספו על ידי המשתמש, וכל מעברי השורה ייעלמו. הטקסט ייראה כשורה אחת ארוכה.

הפונקציה הנראית בתדפיס 29.7 משמשת להצגת טקסט קורות החיים.

**תדפיס 29.7** עיצוב HTML והצגת טקסט פשוט

```
FUNCTION formatText( ByVal theText )
    if INSTR( theText, ">" ) > 0 then
        theText = Replace( theText, "<table", "<x", vbtextcompare )
        theText = Replace( theText, "</table",  "<x", vbtextcompare )
        theText = Replace( theText, "<tr", "<x", vbtextcompare )
        theText = Replace( theText, "</tr", "<x", vbtextcompare )
        theText = Replace( theText, "<td", "<x", vbtextcompare )
        theText = Replace( theText, "</td", "<x", vbtextcompare )
formatText = theText
    else
        theText = Replace( theText, vbCRLF&vbCRLF, "<p>" )
        theText = Replace( theText, vbCRLF, "<br>" )
        formatText = theText
    end if
END FUNCTION
```

פונקציה זו בודקת אם טקסט קורות החיים כולל את התו >. אם הטקסט אכן כולל תו זה, הפונקציה מניחה כי הוא מעוצב באמצעות תגיות HTML. כאן טמונה הסכנה. אם טקסט קורות החיים כולל את התגית <TABLE> של HTML, או את התגיות <TR> או <TD>, הצגת קורות החיים תהרוס את מראהו של הדף Resume Detail. הדבר יקרה מכיון שהדף Resume Detail עצמו מציג את תוכן קורות החיים בטבלת HTML משלו. כדי למנוע את הרס הדף מוחלפת כל אחת מהתגיות המסוכנות הללו.

אם המשתמש הקליד את קורות חייו ללא עיצוב HTML, יש לייחס חשיבות להצגת תוכן קורות החיים בהתאם לעיצוב המקורי שלהם. אם לא תנקוט בצעדי מנע מיוחדים יאבדו כל תווי הרווח והקשות ה-Enter שהקיש המשתמש במקור, וקורות החיים יופיעו כפיסקה אחת ארוכה, ומאוד לא נעימה לקריאה.

הפונקציה formatText מתרגמת את ההקשות על Enter (אשר לא יופיעו ב-HTML) לתגיות HTML למעבר שורה, <BR>, או למעבר פיסקה, <P>. שתי הקשות על Enter מתורגמות לתגית <P>, ואילו הקשה יחידה על Enter מתורגמת לתגית <BR>. שים לב שתהליך זה אינו מציג את העיצוב המקורי בדיוק, אך הוא הטוב ביותר שתוכל לעשות באמצעות HTML פשוט.

**הערה:**

ייתכן ותתהה, מדוע לא נעשה בפונקציה formatText שימוש בתגית <PRE> כדי לשמר את הריווח של הטקסט בקורות החיים. התגית <PRE> הוצגה ככזו המאפשרת לשמר טקסט מעוצב, מכיון שהיא משמרת את כל הריווחים המקוריים. לעומת זאת, חסרונה של התגית <PRE> טמון בכך שהיא עושה את מלאכתה טוב מדי. היא מונעת גלישת שורות. אם משתמש כלשהו הקליד את קורות חייו ברצף אחד ללא enter, הטקסט ייגלל כאשר יוצג בחלון HTML רגיל, אך הוא לא ייגלל שוב כאשר יוצג באמצעות התגית <PRE>.

# חיפוש ברשימת הצעות העבודה

דף ASP של חיפוש הצעת עבודה, Search Job, נכלל בקובץ searchjob.asp, והוא זהה כמעט באופן מוחלט לדף Search Resume. אבל, הדף Search Jobs משמש לחיפוש הצעות העבודה המופיעות בטבלה Jobs, ולא לחיפוש קורות חיים המאוחסנים בטבלה Resume. הדף Search Jobs מופיע בתרשים 29.7.

**תרשים 29.7** דף לחיפוש אחר הצעות עבודה

תוכל להשתמש בדף זה לחיפוש אחר הצעת עבודה, כשהוא ממוין לפי מדינה או ארץ. תוכל גם להגביל את החיפוש שלך כך שיציג רק את סוג המשרה המוצעת (משרה מלאה או זמנית). לסיום, תוכל לחפש אחר הצעת עבודה תוך שימוש במילות מפתח.

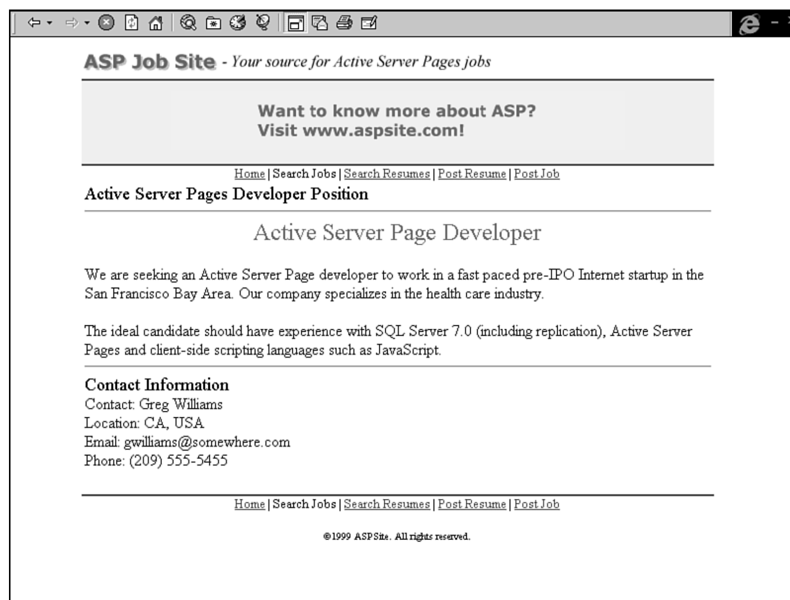
**הערה:**

הצעות עבודה ממשיכות לתפוס נפח דיסק גם לאחר שתוקפן פג. תוכל לבצע מחיקה ידנית של אותם קורות חיים שתוקפם פג מדי תקופת זמן שתקבע לעצמך, או שתוכל ליצור משימת שרת SQL (SQL Server Job) שתבצע זאת באופן אוטומטי.

כאשר המשתמש שולח את הטופס Search Jobs נשלחים הנתונים חזרה לדף Search Jobs עצמו. מחרוזת שאילתה של SQL נבנית מקריטריון החיפוש שהועבר, ותוצאות השאילתה מוצגות. תוצאות החיפוש מופיעות כרשימה של התיאור המקוצר של הצעת העבודה, כאשר כל תיאור מהווה קישור לדף Jobs Detail.

# הצגת הצעת עבודה במלואה

הדף Job Detail נכלל בקובץ jobdetail.asp, ומטרתו היא להציג את כל הנתונים של רשומת הצעת עבודה מסוימת. דף הצגת פרטי המשרה נראה בתרשים 29.8.




**תרשים 29.8** הצגת פרטי המשרה

דף זה פועל בדיוק כמו הדף Resume Detail. כאשר המשתמש לוחץ על קישור מדף תוצאות החיפוש מעביר אותו הקישור לדף המפרט את הצעת העבודה. הקישור כולל משתנה מחרוזת שאילתה בשם id, אשר מזהה באופן ייחודי את הצעת העבודה. ערך משתנה זה תואם לעמודה IDENTITY job_id שבטבלת מסד הנתונים Jobs.

הנתונים לדף Job Detail מאוחזרים מטבלת מסד הנתונים Jobs. השורה הנכונה בטבלה נקראת על ידי התאמה בין המשתנה id לערך העמודה job_id שבשורה.

הטקסט של הצעת העבודה מעוצב באמצעות הפונקציה formatText, אשר מסירה כל מופע של התגיות <TABLE>, <TR> ו-<TD> מתוכן ההצעה. בנוסף, כאמור, היא גם מחליפה את תווי ה-Enter עם תגיות <BR> או <P> של HTML.

# הרחבת הפרויקט

על פי מצב הדברים כעת, האתר ASP Job Site פועל כאתר אינטרנט לכל דבר. תוכל לקחת את אותו קוד עם שינויים קלים בטקסט, וליצור אתר חיפוש עבודה לטבחים, רופאי שיניים ואפילו לטכנאי מקררים. אבל, כל אתר אינטרנט ניתן לשיפור, וסעיף זה מכיל מספר הצעות לאופן בו ניתן להרחיב ולשפר את אתר חיפוש העבודה שלנו.

בעיה אחת עם האתר ASP Job Site שלנו היא שהוא אינו מאפשר למשתמשים בו לעדכן את קורות החיים שלהם או את הצעות העבודה, לאחר שהם פורסמו באתר. למשל, אם מעסיק כלשהו מצא את העובד שחיפש, אין לו שום דרך להסיר את הצעת העבודה שהוא פירסם. הצעת העבודה תמשיך להופיע בדף תוצאות החיפוש עד שיפוג תוקפה.

כדי לאפשר למשתמשים לעדכן את הנתונים שהם פרסמו, חייבים להוסיף לאתר מערכת כניסה עם שם משתמש וסיסמה. תוספת זו הכרחית כדי למנוע ממשתמשים "לא נחמדים" לבצע שינויים לא רצויים בקורות חיים או בהצעות עבודה של משתמשים אחרים.

בעיה נוספת של האתר שלנו, באופן בו הוא מיושם כעת, היא שהוא אינו מאפשר למשתמשים להישאר בעילום שם. אם פירסמת את קורות חייך, עליך לכלול את מספר הטלפון או את כתובת הדואר האלקטרוני שלך, כדי לאפשר למעסיקים אפשריים ליצור עימך קשר. גם המעסיקים חייבים להוסיף להצעות העבודה שלהם באתר את אמצעי הקשר אליהם.

קיימות מספר דרכים בהן תוכל להרחיב את פעילותו של האתר, כדי לאפשר למשתמשים בו להישאר אנונימיים. ראשית, תוכל לאפשר למשתמשים באתר להגיב להצעת עבודה או לקורות חיים. לדוגמה, מעסיק יכול לפרסם הצעת עבודה באופן אנונימי, ואז לדפדף בין התשובות להצעה ולהחליט למי מבין המשיבים להודעה הוא מעוניין להתקשר. התשובות יכולות להיות מאוחסנות בטבלת מסד נתונים.

דרך נוספת לאפשר אנונימיות היא לשלוח אל ומאת המשתמשים הודעות באמצעות CDO עבור CDO for NTS) NTS). לדוגמה. אם מועמד למשרה כלשהי מעוניין ליצור קשר עם מעסיק, הוא יכול להקליד את הודעתו לטופס HTML. מיד לאחר שהטופס נשלח, מועברת ההודעה למעסיק, תוך שימוש ב-CDO עבור NTS למשלוח הודעת דואר אלקטרוני. באופן זה, לא המעסיק ולא מחפש העבודה יראו אחד את כתובת הדואר האלקטרוני של השני. כדי ללמוד אודות השימוש ב-CDO עבור NTS לשליחת דואר אלקטרוני, פנה לפרק 10.

# סיכום

בפרק זה למדת כיצד ליצור אתר חיפוש עבודה מלא ומתפקד, תוך שימוש בתסריטי ASP בלבד. למדת כיצד לאסוף נתונים מטופס HTML ולאחסן אותם בטבלת מסד נתונים. בנוסף למדת כיצד ליצור טפסי חיפוש היכולים לשמש לביצוע שאילתות מותאמות.

# נספחים

לפניך ארבעה נספחים ואינדקס לספר כולו, כולם באנגלית. שים לב, הנספח הראשון מתחיל בעמוד 1 שנמצא בסוף הספר (לפני הקטלוג המצורף) ומתקדם פנימה אל תוך הספר.

לאחר הנספחים הללו תוכל למצוא את האינדקס. הפניות האינדקס הן שתיים: אחת לפי מספרי העמודים של הפרקים בעברית - הם יסומנו במספרים, והשנייה לפי מספרי העמודים של הנספחים - הם יסומנו במספר ובאות E (קיצור ל-English, למשל e18).

# חלק 8

# נספחים
# Appendixes

**In This Part:**

## Appendix A

# Frequently Asked Questions About Active Server Pages

**In This Appendix:**

- Installation Questions
- General Scripting Questions
- Session and Application Variables Questions
- File Questions
- Browser Questions
- ActiveX Data Objects Questions
- Form and Query String Questions
- Questions About Active Server Pages Web Sites

This appendix addresses the questions that are most frequently posted on the Active Server Pages newsgroups and the ASPSite (the companion Web site to this book at www.aspsite.com). When appropriate, references are provided to specific chapters in this book where more information can be found. For updated information, please visit http://www.aspsite.com/answers.

# Installation Questions

## Which Operating Systems Support Active Server Pages?

Active Server Pages runs natively on Microsoft Windows NT Server 4.0, Microsoft Windows NT Workstation 4.0 with Peer Web Services, and Windows 95/98 with the Personal Web Server.

Using Chili!Soft's Chili!ASP (see http://www.chilisoft.com), you also can use Active Server Pages with SUN Solaris and IBM AIX. Chili!ASP enables Active Server Pages to run on Apache servers, Netscape Enterprise and FastTrack servers, the Lotus Domino Go Webserver, and O'Reilly Website Pro.

For more information on installing Active Server Pages, see Chapter 1.

## How Do I Get the Latest Version of Internet Information Server and Active Server Pages?

You can download the latest version of Active Server Pages and Internet Information Server at the Microsoft Web site. Go to http://www.microsoft.com/iis and select Downloads. Download the Windows NT Server 4.0 Option Pack. You will be provided with the option of downloading either the Windows NT or Windows 95/98 version of the Option Pack.

For more information on installing Active Server Pages, see Chapter 1.

## How Do I Get the Latest Version of the ActiveX Data Objects (ADO)?

The latest version of the ActiveX Data Objects (ADO) is available at the Microsoft Web site. Go to http://www.microsoft.com/data and download the Microsoft Data Access Components (MDAC).

For more information on the Microsoft Data Access Components, see Chapter 12.

# General Scripting Questions

## How Do I Add a Quotation Mark to a VBScript String?

There are two ways to add a quotation mark to a VBScript string:

```
myVar = "He said, ""Hello!"""
myVar = "He said, " & CHR( 34 ) & "Hello!" & CHR( 34 )
```

The first method uses two quotation marks in a row to create a single quotation mark. The second method uses the ASCII value of the quotation mark character.

## How Can I Break a Single VBScript Statement into Multiple Lines?

You can break a single VBScript statement into multiple lines by using the _ underscore character. For example, the string in the following statement is broken into several lines of code by using the &_ character combination:

```
myVar = "When in the Course of human events, " &_
   "it becomes necessary for one people to " &_
   "dissolve the political bands which have " &_
   "connected them with another, and to assume " &_
   "among the powers of the earth, the separate " &_
   "and equal station to which the Laws of " &_
   "Nature and of Nature's God entitle them," &_
   "a decent respect to the opinions of " &_
   "mankind requires that they should declare " &_
   "the causes which impel them to the separation. "
```

## What Is the Proper Method of Comparing Strings in VBScript?

When you compare two strings with the identity operator, the comparison is case sensitive. For example, the following statement returns the value false:

```
<%= "apple" = "APPLE" %>
```

There are two methods of performing a case-insensitive comparison of two strings:

```
<%= StrComp( "apple", "APPLE", vbTextCompare )%>
<%= UCASE( "apple" ) = UCASE( "APPLE" ) %>
```

The first method uses the VBScript StrComp function with the vbTextCompare constant. The second method forces both strings into uppercase.

You should be aware that several other VBScript string functions, such as the InStr and Replace functions, are also case sensitive. To perform case-insensitive comparisons with these functions, you must use the vbTextCompare constant.

## How Can I Re-enable Errors After Using ON ERROR RESUME NEXT?

The VBScript ON ERROR RESUME NEXT statement suppresses errors in your script. If you include the statement outside any functions or subroutines, the statement will apply to every statement that follows it (otherwise, it will apply only within the function or subroutine). To re-enable the reporting of errors, use the ON ERROR GOTO 0 statement like this:

```
<%
ON ERROR RESUME NEXT
' The following error is ignored
fakeOBJ.Blah
ON ERROR GOTO 0
' The following error is reported
fakeOBJ2.Blah
%>
```

For more information on detecting errors in your Web site, see Chapter 24.

## How Can I Prevent My Script from Timing Out?

By default, an Active Server Page script will stop executing and time out after 90 seconds. If you have a long running script, the script may stop executing too early. You can extend the amount of time a script is allowed to run by using the ScriptTimeout property of the Server object. Here’s an example:

```
<% Server.ScriptTimeout = 200 %>
```

This statement changes the timeout period to 200 seconds.

## When Do I Need to Explicitly Convert a Variable to a Particular Data Type?

If you are comparing numbers or dates and times and there is a danger that VBScript might interpret the values as strings, you should use one of the VBScript conversion functions.

For example, suppose an HTML form contains two input boxes for two numbers, and you execute the following script:

```
<%
firstNum = Request( "firstNum" )
secondNum = Request( "secondNum" )
%>
<%=firstNum > secondNum %>
```

If you enter 32 for the first number and 223 for the second number into the form, the script will output the wrong result. The script will return the value TRUE because the string 32 is greater than the string 223, even though the number is not. To force an integer comparison, rather than a string comparison, use a script like this:

```
<%
firstNum = CInt( Request( "firstNum" ) )
secondNum = CInt( Request( "secondNum" ) )
%>
<%=firstNum > secondNum %>
```

The CInt function converts a value to the Integer subtype (also see the CDate, CCur, IsNumeric, and IsDate functions).

## When Should I Pass a Variable by Value and When by Reference?

When you pass a variable by value to a subroutine or function, a new instance of the variable is created. Any changes made to the value of the variable do not affect the value of the original variable. On the other hand, when you pass a variable by reference, changes made to the variable do affect the value of the original variable. Here's an example:

```
<%
SUB addOne( ByVal fvar, ByRef svar )
  fvar = fvar + 1
  svar = svar + 1
END SUB
firstvar = 0
secondvar = 0
addOne firstvar, secondvar
%>
```

After this script is executed, the variable named firstvar has the value 0 and the variable named secondvar has the value 1.

## How Can I Add a Space Between the Output of Two Variables?

For some mysterious reason, when you output two variables in a row in an Active Server Page, any spaces that appear between the variables will disappear. For example, the following script outputs "AndrewJones" rather than "Andrew Jones":

```
<%
fname = "Andrew"
lname="Jones"
%>
<%=fname%> <%=lname%>
```

The easiest way to get around this problem is to use a script like the following:

```
<%
fname = "Andrew"
lname="Jones"
%>
<%=fname & " " & lname%>
```

## Does Active Server Pages Support Dynamic Includes?

Some server-side scripting environments, such as Cold Fusion, support dynamic includes. A dynamic include enables you to use a variable for the name of the file to include in a page. Active Server Pages does not support dynamic includes. The following script will not work:

```
<%
myFile = "firstpage.asp"
%>
<!-- #INCLUDE FILE="<%=myFile%>" -->
```

This include directive will attempt to include a file named "<%=myFile%>", which probably does not exist. The problem is that all server-side directives, including the #INCLUDE directive, are processed before Active Server Page scripts. If you must dynamically include different pages within an Active Server Page, consider using a script like the following:

```
<%
myFile = "firstpage.asp"
if myFile = "firstpage.asp" then
%>
<!-- #INCLUDE FILE="firstpage.asp" -->
<%
```

```
end if
if myFile = "secondpage.asp" then
%>
<!-- #INCLUDE FILE="secondpage.asp" -->
<%
end if
if myFile = "thirdpage.asp" then
%>
<!-- #INCLUDE FILE="thirdpage.asp" -->
<%
end if
%>
```

This script conditionally displays one of three different pages. The version of Active Server Pages bundled with IIS 5.0 will include better methods for dynamically including files. See Chapter 26 for more information.

## How Do I Use the Option Explicit Statement in an Active Server Page?

The Option Explicit statement forces you to declare all your variables. Using this statement can make it easier to program complicated Active Server Pages because misspelled variable names will generate errors.

You must use the Option Explicit statement before any other VBScript statement or HTML content in a script. If you don't, you'll receive an error like the following:

```
Microsoft VBScript compilation error '800a0400'

Expected statement
/test.asp, line 5
Option Explicit
^
```

Here's an example that uses the Option Explicit statement correctly:

```
<Option Explicit %>
<html>
<head><title>My Page</title></head>
<body>
<%
DIM myVar
myVar = "Hello!"
%>
</body>
<html>
```

# Session and Application Variables Questions

## Why Do Session Variables Sometimes Fail to Work?

Session variables depend on browser cookies. If a browser does not support cookies, or a user has turned off cookies or the user's cookie file is corrupted, Session variables will not work. For some suggestions on avoiding Session variables, see Chapter 5.

## How Can I Remove an Application Variable?

The current version of Active Server Pages, included with IIS 4.0, does not contain a method for removing Application variables. After you create one, it remains in memory until the Internet Service is stopped, the Global.asa file is changed, or the current application is unloaded.

The version of Active Server Pages included with IIS 5.0, on the other hand, includes two new methods for removing Application variables: the Remove and RemoveAll methods. For more information on IIS 5.0, see Chapter 26.

# File Questions

## How Do I Detect Whether a File Exists?

You can check whether a file exists by using the FileExists method of the FileSystemObject. The following script detects whether a file named test.txt exists:

```
<%
Set fs = Server.CreateObject( "Scripting.FileSystemObject" )
if fs.FileExists( "c:\test.txt" ) then
  Response.Write "File Exists!"
else
  Response.Write "No File!"
end if
%>
```

To learn more about the FileSystemObject see Chapter 8.

## How Can I Automatically Display a List of Files in a Directory?

You can display a list of files in a folder by using the FileSystemObject and the Folder object like this:

```
<%
Set fs = Server.CreateObject( "Scripting.FileSystemObject" )
Set folder = fs.GetFolder( "c:\myfolder" )
For Each thing In folder.Files
  Response.Write thing.name & "<br>"
Next
%>
```

This script lists the names of all the files in a folder named myfolder located on the c: drive. For more information see Chapter 8.

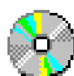
# Image Questions

## How Can I Store an Image in a Database Table?

Although you can store an image in a SQL Server BLOB column, it is almost always better to store the URL of an image in a database table rather than the image itself. (Storing an image in a database table places unnecessary work on your database server.) For example, the following script displays several images in a row by retrieving the URL of each image from a database table named myImages:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
mySQL = "SELECT image_URL from myImages"
Set RS = Con.Execute( mySQL )
While Not RS.EOF
%>
<img src="<%=RS( "image_URL" )%>">
<%
RS.MoveNext
Wend
%>
```

## How Can I Dynamically Generate a Graph or Image in an Active Server Page?



Several third-party Active Server Pages components enable you to dynamically create custom images. One of the best of these components is

the ShotGraph component included on the CD-ROM that accompanies this book. For more information, see Chapter 16.

# Browser Questions

## How Do I Detect the Type of Browser a Person Is Using to Visit My Web Site?

The USER-AGENT header indicates the type of browser. The following script captures the value of the USER-AGENT header from the ServerVariables collection and displays it:

```
<%
browserType = Request.ServerVariables( "HTTP_USER_AGENT" )
Response.WRite browserType
%>
```

For more information on the ServerVariables collection, see Chapter 2.

## How Can I Detect the Page From Which a Person Originated?

The REFERER header contains the URL of the last page the person visited. This header can be retrieved from the ServerVariables collection. The following script displays the value of the REFERER header:

```
<%
browserType = Request.ServerVariables( "HTTP_REFERER" )
Response.WRite browserType
%>
```

For more information on the ServerVariables collection, see Chapter 2.

# ActiveX Data Objects Questions

## Why Do I Sometimes Receive an Error When Using the Connection Object to Execute a SQL String?

Consider the following script. It retrieves a user’s first name from an HTML form and inserts it into a database table:

```
<%
firstname = Request( "firstname" )
mySQL = "INSERT myTable ( firstname ) VALUES "
mySQL = mySQL & "('" & firstname & "')"
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Con.Execute mySQL
%>
```

Suppose, however, that the user entered a single quote when entering a first name. For example, the user entered the name O'Reilly. Because SQL uses a single quote to mark the beginning and end of a string value, the single quote would generate an error. Before entering a string into a database table with the Connection object, you must first translate any single quotes into two quotes in a row. The following script avoids errors caused by a quotation mark:

```
<%
FUNCTION fixQuotes( theVar )
  fixQuotes = REPLACE( theVar, "'", "''" )
END FUNCTION
firstname = Request( "firstname" )
firstname = fixQuotes( firstname )
mySQL = "INSERT myTable ( firstname ) VALUES "
mySQL = mySQL & "('" & firstname & "')"
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Con.Execute mySQL
%>
```

In this example, the fixQuotes function replaces any single quote with two quotes. Doubling the quotes enables you to enter single quotes into a database table.

## How Do I Retrieve a TEXT Field in an Active Server Page?

If you do not take special precautions when retrieving a TEXT field from a database table, the value of the TEXT field might be truncated. If you are using a forward-only cursor type, you should list the TEXT field as the very last field in your select list. Alternatively, consider using a richer cursor type when opening a Recordset that contains a TEXT field. The following script will correctly retrieve and display a TEXT field:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Set RS = Server.CreateObject( "ADODB.RecordSet" )
```

```
RS.CursorType = adOpenDynamic
RS.Open "Select TextColumn FROM mytable", Con
%>
```

This script opens a Recordset with a dynamic cursor. For more information about Recordset cursors see Chapter 13.

## Why Does RecordCount Always Return the Value -1?

The RecordCount property returns the number of rows in a Recordset after it has been opened. You cannot use this property with a forward-only cursor when using SQL Server (it will always return -1). To use this property, open a Recordset with a richer cursor type like this:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
mySQL = "SELECT * FROM WebUsers"
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.CursorType = adOpenDynamic
RS.Open mySQL, Con
Response.Write RS.RecordCount
%>
```

This script opens a Recordset with a dynamic cursor. For more information about Recordset cursors see Chapter 13.

## Why Do I Receive an Error Whenever I Try to Update the Value of a Field in a Recordset?

By default, when you open a Recordset, it is opened with a forward-only cursor and read-only lock type. To update a Recordset, you must open a Recordset that is not read-only. Here is an example:

```
<!-- #INCLUDE VIRTUAL="/adovbs.inc" -->
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.LockType = adLockOptimistic
RS.Open "SELECT  * FROM Webusers WHERE 1<>1", Con
RS.AddNew
RS( "username" ) = "Andrew Jones"
RS( "password" ) = "won't say"
RS.Update
%>
```

This script opens a Recordset with an optimistic locking type and adds a new record to a table named Webusers.

# How Can I Limit the Number of Rows Returned by a Database Query?

If you are using Microsoft SQL 7.0 or Microsoft Access, you can use the SQL TOP keyword to limit the number of records returned. Otherwise, if you are using Microsoft SQL 6.5, use the MaxRecords property of the Recordset object like this:

```
<%
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=pubs"
Set RS = Server.CreateObject( "ADODB.Recordset" )
RS.MaxRecords = 15
RS.Open "SELECT * FROM Authors ORDER BY au_lname", Con
While not RS.EOF
  Response.Write RS( "au_lname" ) & "<br>"
  RS.MoveNext
Wend
%>
```

For more information see Chapter 13.

## How Do I Retrieve the Value from a SQL Statement That Uses COUNT(*), MAX, MIN, or @@IDENTITY?

There are two methods that you can use to retrieve the value from a SQL function or a SQL global variable. You can either use an alias for the value or refer to the value by its ordinal position in the Recordset. The following example displays the value returned from SQL COUNT(*) using both methods:

```
<%
mySQL = "SELECT COUNT(*) theCount from Authors"
Set Con = Server.CreateObject( "ADODB.Connection" )
Con.Open "FILE NAME=c:\myDataLink.UDL;DATABASE=pubs"
SET RS = Con.Execute( mySQL )
Response.Write RS( "theCount" )
Response.Write RS( 0 )
%>
```

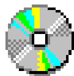
The first Response.Write statement uses the alias theCount. The second Response.Write statement uses the ordinal position of the value.

### How Do I Retrieve an Output Parameter from a SQL Stored Procedure Within an Active Server Page Script?

To retrieve an output parameter or a return code, you must use the ADO Command and Parameter objects. See Chapter 14.

# Form and Query String Questions

### How Can I Accept File Uploads in an HTML Form?



See Chapter 3 for an example of how you can accept and work with file uploads from an HTML form with Active Server Page scripts. See Chapter 16 for a discussion of the Software Artisan's SA-FileUP component, which is included on the CD-ROM that accompanies this book.

### What Causes the Value of a Form Field to Become Truncated?

HTML uses quotation marks to mark the beginning and end of a value. So, if a variable contains quotation marks in its value, the value of the variable will be truncated when displayed. For example, the HTML form in the following script will not display correctly:

```
<%
myVar = "He said, ""Hello!"""
%>
<form method="post" action="page.asp">
<input name="myfield" type="text"
  value="<%=myVar%>">
<input type="submit" value="Enter">
</form>
```

Instead of displaying the string He said, "Hello!" as the value of the form element, the value He said, appears. Every character after and including the " is cut off. To get around this problem, HTML encode the string before displaying it, like this:

```
<%
myVar = Server.HTMLEncode( "He said, ""Hello!""" )
%>
<form method="post" action="page.asp">
<input name="myfield" type="text"
  value="<%=myVar%>">
<input type="submit" value="Enter">
</form>
```

For more information on working with HTML forms, see Chapter 3.

### How Can I Include Spaces or Other Special Characters in a Query String?

To include spaces or other special characters—such as periods and quotation marks—in a query string, URL encode the query string. The following example correctly encodes a query string before displaying it:

```
<%
myVar = Server.URLEncode( "He said, ""Hello!""" )
%>
<a href="page.asp?qvar=<%=myVar%>">Go</a>
```

For more information on using query strings, see Chapter 2.

# Questions About Active Server Pages Web Sites

### Do Any Internet Service Providers Host Active Server Pages?

Several good Internet service providers host sites that use Active Server Pages and SQL Server. Two examples are Bitshop (www.bitshop.com) and Data Return (www.datareturn.com). Both companies enable you to administer an Active Server Page site remotely using Microsoft Visual InterDev and Microsoft SQL Server Enterprise Manager.

### What Good Web Sites Have Information on Active Server Pages?

The number of good Web sites that have information on Active Server Pages is growing quickly. Here is a list of some of the Web sites that I visit on a weekly basis:

| | |
|---|---|
| www.aspsite.com | The companion Web site to this book. |
| www.15seconds.com | This Web site has thousands of pages of information on Active Server Pages. |

| | |
|---|---|
| www.activeserverpages.com | This Web site has a great component section and several interesting articles. |
| www.asphole.com | This Web site contains lists of components and articles related to Active Server Pages. |
| www.swynk.com | Good site for information on Active Server Pages and SQL Server. |
| www.microsoft.com/iis | Microsoft's Internet Information Server site. |

## Appendix B

# ASP Objects and Components Reference

**In This Appendix**

- **ASP Built-In objects**
- **ASP Installable Components**

This appendix provides a quick reference for the Active Server Page objects and installable components. This reference is not intended to be comprehensive. For more information on each object and component, consult the Microsoft online documentation (available at http://localserver/iishelp on a server with IIS installed).

# ASP Built-In Objects

You can use any of the following Active Server Pages built-in objects without explicitly declaring them in your Active Server Page scripts.

## The Application Object

The Application object can be used to store and retrieve information that can be shared by all users of an application. For more information on the Application object, see Chapter 4.

**Collections**

| | |
|---|---|
| Contents(key) | Contains data and objects with application scope not declared with the <object> tag |
| StaticObjects(key) | Contains objects with application scope declared with the <object> tag |

**Methods**

| | |
|---|---|
| Lock | Prevents other users from accessing the Application collection |
| Unlock | Enables other users to access the Application collection |

**Events**

| | |
|---|---|
| OnEnd | Triggered by stopping the Web server, modifying the Global.asa file, or unloading the application |
| OnStart | Triggered by the first request for a Web page in the application |

## The ObjectContext Object

The ObjectContext object is used to control Active Server Page transactions. For more information on the ObjectContext object, see Chapter 21.

**Methods**

| | |
|---|---|
| SetAbort | Explicitly aborts a transaction |
| SetComplete | Overrides any calls to the SetAbort method |

**Events**

| | |
|---|---|
| OnTransactionAbort | Triggered by Aborted transaction |
| OnTransactionCommit | Triggered by Completed transaction |

## The Request Object

The Request object represents all information sent in a request from a browser to your server. For more information on the Request object, see Chapter 2.

**Collections**

| | |
|---|---|
| ClientCertificate(key) | Contains values from client-certificate |
| Cookies(key) | Contains values from browser cookies |
| Form | Contains values from HTML form fields |
| QueryString | Contains values from query string |
| ServerVariables | Contains values from headers and environmental variables |

**Method**

| | |
|---|---|
| BinaryRead(count) | Retrieves raw content of HTML form |

**Property**

| | |
|---|---|
| TotalBytes | The number of bytes in the browser request |

## The Response Object

The Response object represents all information that is sent from your server to a browser. For more information on the Response object see Chapter 2.

**Collection**

| | |
|---|---|
| Cookies(key) | Enables you to add a cookie to a browser |

**Methods**

| | |
|---|---|
| AddHeader name, value | Adds new header to a browser |
| AppendToLog | Adds an entry to the IIS log file |

**Methods**

| | |
|---|---|
| BinaryWrite data | Writes non-string information in the response |
| Clear | Clears the buffered response |
| End | Ends processing of a script |
| Flush | Sends all contents of the buffer |
| Redirect URL | Redirects a browser to a new page |
| Write variant | Sends a string to a browser. |

**Properties**

| | |
|---|---|
| Buffer | Buffers an Active Server Page |
| CacheControl | Controls caching by proxy servers |
| CharSet(Charsetname) | Character set to use on current page |
| ContentType | Specifies the content type of the response |
| Expires | Controls caching by browsers using relative time |
| ExpiresAbsolute | Controls caching by browsers using absolute time |
| IsClientConnected | Indicates whether a browser is still connected |
| PICS(PICSLABEL) | Adds a PICS rating to a page |
| Status | Specifies the status line returned by server |

## The Server Object

The Server object enables the use of various utility functions on the server.

**Methods**

| | |
|---|---|
| CreateObject(progID) | Creates an instance of an object |
| HTMLEncode(string) | Converts a string to use special HTML characters |
| MapPath(path) | Converts a virtual path to a physical path |
| URLEncode(string) | Converts a string to URL-encoded form |

**Property**

| | |
|---|---|
| ScriptTimeout | The number of seconds a script is allowed to run before being terminated |

## The Session Object

The Session object represents the information about a particular user session. For more information on the Session object see Chapter 5.

**Collections**

| | |
|---|---|
| Contents(key) | Contains data and objects with session scope declared without using the <object> tag |
| StaticObjects(key) | Contains objects with session scope declared using the <object> tag |

**Method**

| | |
|---|---|
| Abandon | Ends a user session after the current page is processed |

**Properties**

| | |
|---|---|
| CodePage | Specifies the server code page |
| LCID | Specifies the location identifier |
| SessionID | A unique identifier for the user session |
| Timeout | The time in minutes before a session terminates |

**Events**

| | |
|---|---|
| OnEnd | Triggered by the user not requesting a page longer than the Session Timeout period |
| OnStart | Triggered by the first request for a Web page by the user |

# ASP Installable Components

Before using any of the following installable components in your Active Server Page scripts, you must explicitly declare an instance of the component. A sample declaration is provided for each component.

## The Ad Rotator Component

The Ad Rotator component enables you to display banner advertisements with different frequencies. For more information on the Ad Rotator component see Chapter 9. To create an instance of the Ad Rotator component use

```
<% Set MyAd=Server.CreateObject("MSWC.AdRotator") %>
```

**Method**

| | |
|---|---|
| GetAdvertisement | Displays an advertisement on a page |
| (Rotator Schedule File) | |

**Properties**

| | |
|---|---|
| Border(size) | Displays a border around the advertisement with a size in pixels |
| Clickable(boolean) | Specifies whether an advertisement is a hyperlink |
| TargetFrame(frame) | The HTML frame where the advertisement should be displayed |

**Files**

| | |
|---|---|
| Redirection File | When you click an advertisement, you're sent to this page |
| Rotator Schedule File | Text file that lists the properties of the advertisements |

## The Browser Capabilities Component

The Browser Capabilities component enables you to detect the features of a Web browser. For more information on the Browser Capabilities component see Chapter 7. To create an instance of the Browser Capabilities component use

```
<% Set MyBrow=Server.CreateObject("MSWC.BrowserType") %>
```

File

| | |
|---|---|
| Browscap.ini | Text file that contains information about browsers |

## The Content Linking Component

For more information on the Content Linking component see Chapter 7. To create an instance of the Content Linking component use

```
<% Set myLink=Server.CreateObject("MSWC.NextLink") %>
```

**Methods**

| | |
|---|---|
| GetListCount(Content Linking List File) | Returns total number of pages in the Content Linking List File |
| GetListIndex(Content Linking List File) | Returns the position of the current page in the Content Linking List File |
| GetNextDescription (Content Linking List File) | Returns the description of the next page in the Content Linking List File |
| GetNextURL(Content Linking List File) | Returns the URL of the next page in the Content Linking List File |
| GetNthDescription (Content Linking List File, index) | Returns the description of the page with the index in the Content Linking List File |
| GetNthURL (Content Linking List File, index) | Returns the URL of the page with the index in the Content Linking List File |
| GetPreviousDescription (Content Linking List File) | Returns the description of the previous page in the Content Linking List File |
| GetPreviousURL (Content Linking List File) | Returns the URL of the previous page in the Content Linking List File |

**File**

| | |
|---|---|
| Content Linking List File | Text file containing a list of pages |

## The Content Rotator Component

The Content Rotator component enables you to display HTML strings with different frequencies. For more information on the Content Rotator component see Chapter 7. To create an instance of the Content Rotator component use

```
<% Set myContent=Server.CreateObject("IISSample.ContentRotator") %>
```

**Methods**

| | |
|---|---|
| ChooseContent(Content Schedule File) | Returns an HTML string from the Content Schedule File |
| GetAllContent(Content Schedule File) | Returns all HTML strings from the Content Schedule File |

**File**

| | |
|---|---|
| Content Schedule File | Text file that contains a list of HTML strings |

## The Counters Component

The Counters component enables you to create one or more counters to track information such as the number of visits to a Web site. For more information on the Counters component see Chapter 9. To create an instance of the Counters Component include the following HTML code in the Global.asa file:

```
<OBJECT RUNAT=SERVER SCOPE=Application ID=myCounter
PROGID="MSWC.Counters"></OBJECT>
```

**Methods**

| | |
|---|---|
| Get(counter) | Returns the value of a counter |
| Increment(counter) | Adds one to the value of a counter |
| Remove(counter) | Removes a counter |
| Set counter, number | Sets counter to equal a certain numeric value |

**File**

| | |
|---|---|
| Counters.txt | This text file contains all the individual counters |

## The Lookup Table Component

The Lookup Table component enables you to cache information in an object that can be created with page, session, or application scope. For more information on the Lookup Table component see Chapter 4. To create an instance of the Lookup Table component with page scope use the following:

```
<% SET myLookup = Server.CreateObject( "IISSample.LookupTable" )%>
```

To create an instance of the Lookup Table component with application scope add the following code to the Global.asa file:

```
<OBJECT RUNAT=SERVER SCOPE=Application ID=myLookup
PROGID="IISSample.LookupTable"></OBJECT>
```

**Methods**

| | |
|---|---|
| Count | Returns the number of key/value pairs that exist in the Lookup table. |
| LoadValuesfilepath, format | Loads key/value pairs into the component from a file. See the values for format. |
| LoadValuesExfilepath, format, period | Loads key/value pairs into the component from a file on a periodic basis. See the values for format. The value for period is specified in seconds. |

**Methods**

| | |
|---|---|
| LookupValue(key) | Returns the value for a specific key. |
| Key(index) | Returns the key located at the index position. |
| KeyExists(key) | Returns True if the specified key exists in the Lookup Table. |
| ReadLock | Locks the Lookup Table until the ReadUnlock method is called. |
| ReadUnlock | Releases a lock created by calling ReadLock. |
| Value(index) | Returns the value of the key located at the index position. |

**File**

| | |
|---|---|
| Lookup Table File | Text file that stores the key/value pairs. Each key/value pair is separated by a comma. To create multiline values, use \ at the end of a line as a continuation character. |

**Format**

| | |
|---|---|
| 0 | String keys and string values. |
| 1 | String keys and integer values. |
| 2 | Integer keys and string values. |
| 3 | Integer keys and integer values. |
| 10 | String keys and string values (ignore duplicate keys). |
| 11 | String keys and integer values (ignore duplicate keys). |
| 12 | Integer keys and string values (ignore duplicate keys). |
| 13 | Integer keys and integer values (ignore duplicate keys). |

## The NewMail Object

The NewMail object enables you to send email from an Active Server Page. For more information on the NewMail object see Chapter 10. To create an instance of the NewMail object use

```
<%SET myMail = Server.CreateObject("CDONTS.Newmail")%>
```

**Methods**

| | |
|---|---|
| AttachFile(file path) | Attaches a file to an email message |
| AttachURL(file path, URL) | Attaches a file to an email message with an associated URL |
| Send | Sends an email message |

**Properties**

| | |
|---|---|
| Bcc | The email address used for sending a blind carbon copy of the email message |
| Body | The text of the email message |
| BodyFormat | Specifies whether the email message contains HTML or plain text |
| Cc | The email address used for sending a carbon copy of the email message |
| ContentBase | Sets the base URL for all URLs included in the email message |
| ContentLocation | Sets an absolute or relative path for all URLs included in the email message |
| From | The email address of the sender of the email message |
| Importance | The priority of the email message |
| MailFormat | The encoding format of the email message |
| Subject | The subject line of the email message |
| To | The email address of the recipient of the email message |

## The Page Counter Component

The Page Counter component enables you to track the number of times that a page has been requested. For more information on the Page Counter component see Chapter 9. To create an instance of the Page Counter component use

```
<% Set myCNT=Server.CreateObject("IISSample.PageCounter")%>
```

**Methods**

| | |
|---|---|
| Hits([Web Page Path]) | Returns the number of times a page has been requested |
| PageHit() | Updates the number of times the current page has been requested |
| Reset([Web Page Path]) | Resets to zero the number of times a page has been requested |

**File**

| | |
|---|---|
| Hit Count Data File | Text file that stores the number of requests that each page has received |

## Appendix C

# VBScript Reference

**In This Appendix:**

- **Statements**
- **Functions**
- **Operators**
- **Objects**
- **Constants**

This appendix contains a quick reference of all the VBScript statements, functions, operators, and objects. Many of the VBScript functions use constants. See the end of this appendix for a list of these constants.

For more information on VBScript, consult the Microsoft VBScript online documentation included with the Windows NT Option Pack or visit the Microsoft Windows Script Technologies Web site at http://msdn.microsoft.com/scripting.

# Statements

## Call

[Call] *name* [*argumentlist*]

This statement transfers control to a function or subroutine. Using Call when calling a function or subroutine is always optional. However, if the optional Call keyword is used, *argumentlist* must be enclosed in parentheses.

## Class

Class *name*
[*statements*]
End Class

Defines a class, assigning a name, variables, properties, and methods.

## Const

[Public | Private] Const *constantname=expression*

This statement is used to declare a constant. You can declare multiple constants within a single line by separating each constant assignment with a comma.

## Dim

Dim *varname*[([*subscripts*])][, *varname*[([*subscripts*])]] . . .

This statement creates a new variable and allocates storage space.

## Do...Loop

Syntax 1:
Do [{While | Until} *condition*]
[*statements*]
[Exit Do]
[*statements*]
Loop

Syntax 2:
Do
[*statements*]
[Exit Do]
[*statements*]
Loop [{While | Until} *condition*]

Both forms of the statement repeat *statements* while *condition* is TRUE or until it becomes TRUE.

## Erase

Erase *array*

This statement erases *array*, reinitializing elements of fixed-size arrays and recovering storage space of dynamic arrays.

## Execute

Execute *statement*

Executes a statement at runtime. The statement can contain multiple statements divided by colons.

## Exit

Exit Do
This statement exits a Do...Loop statement.

Exit For
This statement exits a For...Next or For Each...Next loop.

Exit Function
This statement exits a function.

Exit Sub
This statement exits a subroutine.

## For...Next

For *counter* = *start* To *end* [Step *step*]
[*statements*]
[Exit For]
[*statements*]
Next

This loop repeats a group of *statements* the number of times designated by the loop counter.

## For Each...Next

For Each *element* In *group*
[*statements*]
[Exit For]
[*statements*]
Next [*element*]

For each element in the array or collection, this loop repeats the group of *statements*.

## Function

[Public | Private] Function *name* [(*arglist*)]
[*statements*]
[name = expression]
[Exit Function]
[*statements*]
[*name* = *expression*]
End Function

This statement defines a Function, assigning a name, arguments, and code.

## If...Then...Else

Syntax 1:

If *condition* Then *statements* [Else *elsestatements*]

Syntax 2:

If *condition* Then
*statements*
[ElseIf *condition-n* Then
[*elseifstatements*]] . . .
[Else
[*elsestatements*]]
End If

Both forms of the statement conditionally execute groups of statements.

## On Error

On Error Resume Next

When an error occurs, this statement executes the statement immediately following the statement that caused the runtime error, or executes the statement immediately

following the most recent call out of the procedure containing the On Error Resume Next statement. To re-enable the reporting of errors, use On Error Goto 0.

## Option Explicit

Option Explicit

This statement forces the explicit declaration of all variables using the Dim, Private, Public, or ReDim statements. It must appear before any other statement or content in the Active Server Page.

## Private

Private *varname*[([*subscripts*])][, *varname*[([*subscripts*])]] . . .

This statement creates private variables (variables available only to the script in which they were declared) and allocates storage space.

## Property Get

[Public [Default]| Private] Property Get *name* [(*arglist*)]
[*statements*]
[[Set] *name* = *expression*]
[Exit Property]
[*statements*]
[[Set] *name* = *expression*]
End Property

Defines a property procedure that returns the value of a property for a class.

## Property Let

[Public | Private] Property Let *name* ([*arglist,*] value)
[*statements*]
[Exit Property]
[*statements*]
End Property

Defines a property procedure that sets the value of a property for a class.

## Property Set

[Public | Private] Property Set *name*([*arglist,*] *reference*)
[*statements*]
[Exit Property]
[*statements*]
End Property

Defines a property procedure that sets a reference to an object for a class.

## Public

Public *varname*[([*subscripts*])][, *varname*[([*subscripts*])]] . . .

This statement creates public variables (variables available to all procedures in all scripts in all projects) and allocates storage space.

## Randomize

Randomize [*number*]

This statement gives the Rnd function's random-number generator a new seed value. If you use this statement without supplying a number, the system timer is used for the seed value. If you do not use Randomize before using the RND() function, the same sequence of numbers will be generated every time RND() is called.

## ReDim

ReDim [Preserve] *varname*(*subscripts*) [, *varname*(*subscripts*)] . . .

This statement revises dimension subscripts, sizing or resizing a dynamic array. Preserve protects data in the existing array.

## Rem

Syntax 1:
Rem *comment*

Syntax 2:
' comment

Both forms of this statement keep comments from being parsed. If Rem follows other statements on a line, it must be separated by a colon.

## Select Case

Select Case *testexpression*
[Case *expressionlist-n*
[*statements-n*]] . . .
[Case Else *expressionlist-n*
[*elsestatements-n*]]
End Select

This statement executes statements paired with any *expressionlist* that matches testexpression. If *testexpression* doesn’t match any *expressionlist*, the statements paired with Case Else are executed.

### Set

Set *objectvar* = {*objectexpression* | Nothing}

This statement sets the object reference for the variable or the property. Nothing dissociates *objectvar* from any specific object.

### Sub

[Public | Private] Sub *name* [(*arglist*)]
[*statements*]
[Exit Sub]
[*statements*]
End Sub

This statement defines a subroutine, assigning a name, arguments, and code.

### While...Wend

While *condition*
[*statements*]
Wend

This loop continues to execute the series of *statements* as long as *condition* is TRUE.

### With

With *object*
[*statements*]
End With

With enables you to perform a series of statements with a single object without explicitly referring to the object in every statement.

# Functions

### Abs(*number*)

This function returns an absolute value.

### Array(*arglist*)

This function creates an array.

### Asc(*string*)

This function returns the ANSI code for the first letter in *string*.

## **AscB(*string*)**

This function returns the first byte in *string*.

## **AscW(*string*)**

This function returns the first Unicode character code in *string*.

## **Atn(*number*)**

This function returns the arctangent.

## **CBool(*expression*)**

This function converts to variant of subtype Boolean.

## **CByte(*expression*)**

This function converts to variant of subtype Byte.

## **CCur(*expression*)**

This function converts to variant of subtype Currency.

## **CDate(*date*)**

This function converts to variant of subtype Date.

## **CDbl(*expression*)**

This function converts to variant of subtype Double.

## **Chr(*charcode*)**

This function converts the ANSI character code to the corresponding keyboard character.

## **ChrB(*charcode*)**

This function converts the character code to a single byte.

## **ChrW(*charcode*)**

This function converts the Unicode character code to the corresponding keyboard character.

## CInt(*expression*)

This function converts to variant of subtype Integer.

## CLng(*expression*)

This function converts to variant of subtype Long.

## Cos(*number*)

This function returns the cosine.

## CreateObject(*servername.typename*)

This function creates an Automation object. Within an Active Server Page, use the CreateObject method of the Server object instead of this function.

## CSng(*expression*)

This function converts to variant of subtype Single.

## CStr(*expression*)

This function converts to variant of subtype String.

## Date

This function returns the date according to the system.

## DateAdd(*interval,number,date*)

This function adds the time interval to date. The interval argument accepts the following values:

| *Setting* | *Description* |
|---|---|
| yyyy | Year |
| q | Quarter |
| m | Month |
| y | Day of Year |
| d | Day |
| w | Weekday |
| ww | Week of Year |
| h | Hour |
| n | Minute |
| s | Second |

## DateDiff(*interval,date1,date2*[, *firstdayofweek*[,*firstweekofyear*]])

This function returns the number of intervals between two dates. See the later section "Date and Time Constants" for values of *firstdayofweek* and *firstweekofyear*. The *interval* argument accepts the values shown in the following table:

| *Setting* | *Description* |
|---|---|
| yyyy | Year |
| q | Quarter |
| m | Month |
| y | Day of year |
| d | Day |
| w | Weekday |
| ww | Week of year |
| h | Hour |
| n | Minute |
| s | Second |

## DatePart(*interval,date*[,*firstdayofweek*[, *firstweekofyear*]])

This function returns the designated part of the date. See the later section "Date and Time Constants" for values of *firstdayofweek* and *firstweekofyear*. The *interval* argument accepts the values shown in the following table:

| *Setting* | *Description* |
|---|---|
| yyyy | Year |
| q | Quarter |
| m | Month |
| y | Day of year |
| d | Day |
| w | Weekday |
| ww | Week of year |
| h | Hour |
| n | Minute |
| s | Second |

### DateSerial(*year,month,day*)

This function converts to variant of subtype Date.

### DateValue(*date*)

This function converts to variant of subtype Date.

### Day(*date*)

This function returns a number representing the day of the month according to the argument.

### Exp(*number*)

This function raises e to the power of *number*.

### Eval(*expression*)

This function evaluates the expression and returns the result.

### Filter(*InputStrings,Value*[,*Include*[, *Compare*]])

This function creates a new array according to the filter criteria. See the later section "Comparison Constants" for values of *Compare*. *InputStrings* must be a one-

dimensional array of strings. Value is the string to search for.

### Fix(*number*)

This function converts to integer (for a negative number, this function rounds the number up).

### FormatCurrency(*Expression*[,*NumDigitsAfter Decimal*[,*IncludeLeadingDigit*[,*UseParensForNegativeNumbers*[,*GroupDigits*]]]])

This function formats an expression as currency. See the later section "Tristate Constants" for values of *IncludeLeadingDigit*, *UseParensForNegativeNumbers*, and *GroupDigits*.

## FormatDateTime(*Date*[,*NamedFormat*])

This function formats dates and times. See the later section "Date Format Constants" for values of *NamedFormat*.

## FormatNumber(*Expression*[,*NumDigitsAfterDecimal*[,*IncludeLeadingDigit*[,*UseParensForNegativeNumbers*[,*GroupDigits*]]]])

This function formats numbers. See the later section "Tristate Constants" for values of *IncludeLeadingDigit*, *UseParensForNegativeNumbers*, and *GroupDigits*.

## FormatPercent(*Expression*[,*NumDigitsAfterDecimal*[,*IncludeLeadingDigit*[,*UseParensForNegativeNumbers*[,*GroupDigits*]]])

This function formats percentages. See the later section "Tristate Constants" for values of *IncludeLeadingDigit*, *UseParensForNegativeNumbers*, and *GroupDigits*.

## GetLocale

This function returns the current locale (as a 32-bit integer) without setting a new one. The locale determines the formatting of currencies and dates.

## GetObject([*pathname*][,*class*])

This function returns the specified Automation object from the specified file.

## Hex(*number*)

This function returns the hexadecimal value of *number*.

## Hour(*time*)

This function returns the hour according to the *time* argument.

## InputBox(*prompt*[,*title*][,*default*][,*xpos*][,*ypos*][,*helpfile*,*context*])

This function prompts and returns user input. Don’t use this with Active Server Pages!

## InStr([*start*,]*string1*,*string2*[,*compare*])

This function returns the first appearance of *string2* within *string1*. See the later section "Comparison Constants" for values of *compare*.

## InStrB([*start*,]*string1*,*string2*[,*compare*])

This function returns the byte position of the first appearance of *string2* within *string1*. See the later section "Comparison Constants" for values of *compare*.

## InStrRev(*string1*,*string2*[*start*[,*compare*]])

This function returns the first appearance of *string2* within *string1* starting from the end of the string. See the later section "Comparison Constants" for values of *compare*.

## Int(*number*)

This function returns an integer (for a negative number, it rounds the number down).

## IsArray(*varname*)

This function determines whether the variable is an array.

## IsDate(*expression*)

This function determines whether *expression* can be converted to date format.

## IsEmpty(*expression*)

This function determines whether the variable has been initialized.

## IsNull(*expression*)

This function determines whether *expression* is null.

## IsNumeric(*expression*)

This function determines whether *expression* is a number.

## IsObject(*expression*)

This function determines whether *expression* is an Automation object.

## **Join(*list*[,*delimiter*])**

This function joins substrings in an array separated by the character indicated by *delimiter*. It's the opposite of the Split function.

## **LBound(*arrayname*[,*dimension*])**

This function returns the lower limit of the array dimension. It always returns zero with the current version of VBScript.

## **LCase(*string*)**

This function converts a string to lowercase.

## **Left(*string*,*length*)**

This function returns the left string portion of the designated length.

## **LeftB(*string*, *bytes*)**

This function returns the left string portion of the designated number of bytes.

## **Len(*string* | *varname*)**

This function returns the length of the string or the byte size of the variable.

## **LenB(*string*)**

This function returns the number of bytes used to represent the string.

## **LoadPicture(*picturename*)**

This function loads a picture object.

## **Log(*number*)**

This function returns the natural logarithm of the number.

## **LTrim(*string*)**

This function removes extra leading spaces.

## **Mid(*string*,*start*[,*length*])**

This function returns a string portion of the designated length.

### MidB(*string*,*start*[,*bytes*])

This function returns a string portion of the designated number of bytes.

### Minute(*time*)

This function returns the minute according to the *time* argument.

### Month(*date*)

This function returns the month represented by the number.

### MonthName(*month*[,*abbreviate*])

This function returns the month represented by the name.

### MsgBox(*prompt*[,*buttons*][,*title*][*helpfile*, *context*])

This function prompts the user to choose a button and indicates which button the user has chosen. See the later section "MsgBox Constants" for *button* and return values. Don't use this in an Active Server Page!

### Now

This function returns the current date and time according to the system.

### Oct(*number*)

This function returns the octal value of *number*.

### Replace(*expression*,*find*,*replacewith*[,*start*[,*count*[, *compare*]]])

This function replaces the designated substring *find* with the substring *replacewith* the designated number of times. See the later section "Comparison Constants" for values of *compare*.

### Right(*string*,*length*)

This function returns the right string portion of the designated length.

## RightB(*string,bytes*)

This function returns the right string portion of the designated number of bytes.

## Rnd([*number*)])

This function generates a pseudo-random number.

## Round(*number*[,*numdecimalplaces*])

This function rounds *number*.

## RTrim(*string*)

This function removes extra right spaces.

## ScriptEngine

This function returns the name of the scripting language in use.

## ScriptEngineBuildVersion

This function returns the name of the script engine in use.

## ScriptEngineMajorVersion

This function returns the major version number of the script engine in use.

## Second(*time*)

This function returns the second of the minute according to the *time* argument.

## SetLocale(*lcid*)

This function returns the current locale and sets the new locale specified by *lcid*. After the locale is set, currencies and dates are formatted according to the locale.

## Sgn(*number*)

This function returns the sign of *number*.

## Sin(*number*)

This function returns the sine of *number*.

### Space(*number*)

This function creates a string with the specified number of spaces.

### Split(*expression*[,*delimiter*[,*count*[, *compare*]]])

This function splits a string and converts it into an array. It's the opposite of the Join function. See the later section "Comparison Constants" for values of *compare*.

### Sqr(*number*)

This function returns the square root of the specified number.

### StrComp(*string1*,*string2*[,*compare*])

This function performs a string comparison. See the later section "Comparison Constants" for values of *compare*.

### StrReverse(*string1*)

This function reverses the characters of a string.

### String(*number*,*character*)

This function creates a string with character repeated the specified number of times.

### Tan(*number*)

This function returns the tangent of *number*.

### Time

This function returns the current time according to the system.

### TimeSerial(*hour*,*minute*,*second*)

This function returns Date Variant.

### TimeValue(*time*)

This function returns Date Variant containing time.

## Trim(*string*)

This function removes extra spaces at left and right.

## TypeName(*varname*)

This function returns the subtype by name. See the later section "VarType Constants" for return values.

## UBound(*arrayname*[,*dimension*])

This function returns the upper bound of the array dimension. If no dimension is specified, the first dimension is assumed.

## UCase(*string*)

This function converts the string to uppercase.

## VarType(*varname*)

This function returns the subtype by value. See the later section "VarType Constants" for return values.

## Weekday(*date*,[*firstdayofweek*])

This function returns the day of the week by number. See the later section "Date and Time Constants" for values of *firstdayofweek*.

## WeekDayName(*weekday*,*abbreviate*, *firstdayofweek*)

This function returns the day of the week by name. See the later section "Date and Time Constants" for values of *firstdayofweek*.

## Year(*date*)

This function returns the year according to the argument.

# Operators

## + Operator

*result* = *expression1* + *expression2*

Sum. This operator can also be used for string concatenation but & is less ambiguous.

## And Operator

*result* = *expression1* And *expression2*

Logical conjunction. This operator also performs bitwise comparison, returning digit 1 only where both expressions have digit 1.

## & Operator

*result* = *expression1* & *expression2*

String concatenation.

## / Operator

*result* = *number1*/*number2*

This operator divides two numbers and returns a floating-point number.

## Eqv Operator

*result* = *expression1* Eqv *expression2*

Logical equivalence. This operator also performs bitwise comparison, returning digit 1 only where bits in the two expressions are identical.

## ^ Operator

*result* = *number*^*exponent*

This operator raises a number to the power of exponent.

## Imp Operator

*result = expression1* Imp *expression2*

Material implication. This operator also performs bitwise comparison. The following table illustrates logical implication:

| *expression1* | *expression2* | *result* |
|---|---|---|
| TRUE | TRUE | TRUE |
| TRUE | FALSE | FALSE |
| TRUE | NULL | NULL |
| FALSE | TRUE | TRUE |
| FALSE | FALSE | TRUE |
| FALSE | NULL | TRUE |
| NULL | TRUE | TRUE |
| NULL | FALSE | NULL |
| NULL | NULL | NULL |

The following table illustrates the bitwise comparison associated with the Imp operator:

| *expression1* | *expression2* | *result* |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## \ Operator

result = number1\number2

This operator divides two numbers and returns an integer.

## Is Operator

*result = object1* Is *object2*

This operator checks whether two variables refer to the same object and returns TRUE or FALSE.

## Mod Operator

*result = number1* Mod *number2*

This operator divides two numbers and returns only the remainder.

## * Operator

*result* = *number1* * *number2*

Multiplication.

## - Operator

*result* = *number1*-*number2*

Subtraction.

-number

Sign of number.

## Not Operator

*result* = Not *expression*

Logical negation. This operator also performs bitwise negation.

## Or Operator

*result* = *expression1* Or *expression2*

Logical disjunction. *result* is TRUE if either expression is TRUE or both expressions are TRUE. The Or operator also performs bitwise comparison, returning digit 0 only where both expressions have the digit 0. Elsewhere the digit 1 is returned.

## Xor Operator

*result* = *expression1* Xor *expression2*

Logical exclusion. *result* is TRUE if one and only one expression is TRUE. The Xor operator also performs bitwise comparison, returning the digit 1 only where one and only one expression has the digit 1. Elsewhere the digit 0 is returned.

# Objects

## The Dictionary Object

The Dictionary object can be used to store key/value pairs. To use the Dictionary object in an Active Server Page script, you must declare an instance of the object like this:

```
<% Set myDict = Server.CreateObject( "Scripting.Dictionary" )%>
```

## Methods

### Add *Key*, *Item*

This method adds *Key* and associated *Item* to a Dictionary object.

### Exists(*Key*)

This method checks whether the specified *Key* already exists in the Dictionary object and returns either TRUE or FALSE.

### *Items*

This method returns all *Items* in the Dictionary object as an array.

### *Keys*

This method returns all existing *Keys* in the Dictionary object as an array.

### Remove(*Key*)

This method removes *Key* and its associated item from the Dictionary object.

### RemoveAll

This method removes all keys and their associated items from the Dictionary object.

## Properties

### CompareMode

This property specifies how items in the Dictionary should be compared. See the later section "Comparison Constants" for possible values.

### Count

This property counts items in the Dictionary object (read-only).

### Item(*Key*)

This property returns the item associated with the designated *Key* in the Dictionary object or associates a new value with the key.

### Key(*Key*)

This property sets the specified *Key* in the Dictionary object.

## The Err Object

The Err object contains information about runtime errors and enables you to raise custom errors. You do not need to create an instance of this object to use it in your scripts.

# Methods

### Clear

This method explicitly clears the Err object of all property settings.

### Raise(*number*, *source*, *description*, *helpfile*, *helpcontext*)

This method returns a runtime error. number identifies the type of error. VBScript errors are numbered in the range 0–65535. *source* indicates the object or application that originally generated the error.

# Properties

### Description

This property returns or sets a brief description of an error.

### HelpContext

This property sets or returns the identifier for a topic within the Help file that's appropriate for the Err object.

### HelpFile

This property sets or returns the fully qualified path to the Help file that's appropriate for the Err object.

### Number

This property returns a number identifying the error or, if *errornumber* is included, associates the error with either a VBScript error number or an SCODE error value.

### Source

This property identifies the source of the error, usually by class name or programmatic ID of the object or the application that generated the error.

# The Match Object

The Match object represents the properties of a match that results from calling the Execute method of the RegExp object. You never explicitly create an instance of the Match object.

# Properties

### FirstIndex

This property returns the position where the first match occurs in the string passed to the Execute method of the RegExp object.

### Length

This property returns the length of the match.

### Value

This property returns the actual value of the match.

## The Matches Collection

The Matches collection contains all the Match objects returned by calling the Execute method of the RegExp object.

## Properties

### Count

This property returns the number of Match objects in the Matches collection.

### Item(*key*)

This property returns the Match object specified by *key*.

## The RegExp Object

The RegExp object enables you to perform regular expression pattern matching. To use the RegExp object in an Active Server Page, you must declare an instance of it like this:

```
<% Set myReg = Server.CreateObject( "VBScript.RegExp" ) %>
```

## Methods

### Execute(*string*)

This method performs pattern matching against the *string* using the pattern set with the Pattern property.

### Replace(*string1*, *string2*)

This method replaces the text in *string1* with the text in *string2* according to the pattern set with the Pattern property.

### Test(*string*)

This method returns TRUE if *string* matches the pattern set with the Pattern property.

## Properties

### Global

If this property is assigned the value TRUE, every match is returned. Otherwise, only the first match is returned.

## IgnoreCase

If this property is assigned the value TRUE, then a case-insensitive pattern match is performed.

## Pattern

This property is used to set or return the regular expression pattern that is used when performing matches. The pattern can contain any of the regular expression characters in the following table:

| Character | Description |
|---|---|
| \ | Matches special characters such as the newline character (\n) or escapes regular expression character (\\) |
| ^ | Matches the beginning of input |
| $ | Matches the end of input |
| * | Matches the character that immediately precedes it if it appears zero or more times |
| + | Matches the character that immediately precedes it if it appears one or more times |
| ? | Matches the character that immediately precedes it if it appears zero times or one time |
| . | Matches any single character except the newline character |
| (pattern) | Matches the pattern in the parentheses and remembers the match in the Matches collection |
| x\|y | Matches either character *x* or character *y* |
| {n} | Matches the character that immediately precedes it if it appears exactly *n* times |
| {n,m} | Matches the character that immediately precedes it if it appears at least *n* times and at most *m* times |
| [xyz] | Matches any one of the characters in the brackets |
| [^xyz] | Matches any character not in the brackets |
| [a-z] | Matches any character in the range of characters |
| [^a-z] | Matches any character not in the range of characters |
| \b | Matches a word boundary |

| Character | Description |
|---|---|
| \B | Matches a nonword boundary |
| \d | Matches a numeral |
| \D | Matches any non-numeric character |
| \f | Matches form feed character |
| \n | Matches newline character |
| \r | Matches carriage return character |
| \s | Matches whitespace |
| \S | Matches nonwhitespace character |
| \t | Matches a tab character |
| \v | Matches a vertical tab character |
| \w | Matches any word character or underscore |
| \W | Matches any nonword character |
| \(num) | Matches previously remembered pattern if it appears *num* times |
| \n | Matches ASCII character (*n* represents an ASCII character as an octal escape value) |
| \xn | Matches ASCII character (*n* represents an ASCII character as a hexadecimal escape value) |

# Constants

In the following sections, the constants (left column of each table) can be used in place of the values (in the center column).

## Color Constants

| Constant | Value | Description |
|---|---|---|
| vbBlack | &h00 | Black |
| vbRed | &hFF | Red |
| vbGreen | &hFF00 | Green |
| vbYellow | &hFFFF | Yellow |
| vbBlue | &hFF0000 | Blue |
| vbMagenta | &hFF00FF | Magenta |
| vbCyan | &hFFFF00 | Cyan |
| vbWhite | &hFFFFFF | White |

# Comparison Constants

| **Constant** | **Value** | **Description** |
|---|---|---|
| vbBinaryCompare | 0 | Binary comparison |
| vbTextCompare | 1 | Textual comparison |
| vbDatabaseCompare | 2 | Comparison based on information in the database |

# Date and Time Constants

| **Constant** | **Value** | **Description** |
|---|---|---|
| vbSunday | 1 | Sunday |
| vbMonday | 2 | Monday |
| vbTuesday | 3 | Tuesday |
| vbWednesday | 4 | Wednesday |
| vbThursday | 5 | Thursday |
| vbFriday | 6 | Friday |
| vbSaturday | 7 | Saturday |
| vbFirstJan1 | 1 | Week of Jan 1 (default) |
| vbFirstFourDays | 2 | First week of the year that has at least four days |
| vbFirstFullWeek | 3 | First full week of the year |
| vbUseSystem | 0 | Use the date format of the computer's regional settings |
| vbUseSystemDayOfWeek | 0 | Use the first day of the week according to the system settings |

# Date Format Constants

| *Constant* | *Value* | *Description* |
|---|---|---|
| vbGeneralDate | 0 | Display the date and/or time according to the system settings |
| vbLongDate | 1 | Display the date in long date format |
| vbShortDate | 2 | Display the date in short date format |
| vbLongTime | 3 | Display the time in long time format |
| vbShortTime | 4 | Display the time in short time format |

# MsgBox Constants

These constants are used with the MsgBox function to specify buttons and icons displayed in the message box and to identify the default icon.

| Constant | Value | Description |
|---|---|---|
| vbOKOnly | 0 | Show only the OK button. |
| vbOKCancel | 1 | Show OK and Cancel buttons. |
| vbAbortRetryIgnore | 2 | Show Abort, Retry, and Ignore buttons. |
| vbYesNoCancel | 3 | Show Yes, No, and Cancel buttons. |
| vbYesNo | 4 | Show Yes and No buttons. |
| vbRetryCancel | 5 | Show Retry and Cancel buttons. |
| vbCritical | 16 | Show Critical Message icon. |
| vbQuestion | 32 | Show Warning Query icon. |
| vbExclamation | 48 | Show Warning Message icon. |
| vbInformation | 64 | Show Information Message icon. |
| vbDefaultButton1 | 0 | The first button is the default. |
| vbDefaultButton2 | 256 | The second button is the default. |
| vbDefaultButton3 | 512 | The third button is the default. |
| vbDefaultButton4 | 768 | The fourth button is the default. |

The constants in the following table specify modality:

| Constant | Value | Description |
|---|---|---|
| vbApplicationModal | 0 | The current application won't continue until the user responds to the message box. |
| vbSystemModal | 4096 | No applications will continue until the user responds to the message box. |

The constants in the following table identify which button has been pressed:

| Constant | Value | Description |
|---|---|---|
| vbOK | 1 | OK button |
| vbCancel | 2 | Cancel button |
| vbAbort | 3 | Abort button |
| vbRetry | 4 | Retry button |
| vbIgnore | 5 | Ignore button |
| vbYes | 6 | Yes button |
| vbNo | 7 | No button |

## String Constants

| Constant | Value | Description |
|---|---|---|
| vbCr | Chr(13) | Carriage return |
| vbCrLf | Chr(13) & Chr(10) | Combination carriage return and line feed |
| vbFormFeed | Chr(12) | Form feed |
| vbLF | Chr(10) | Line feed |
| vbNewLine | Chr(13)&Chr(10) or Chr (10) | Newline character appropriate for platform |
| vbNullChar | Chr(0) | Character of value 0 |
| vbNullString | string having value 0 | Null string |
| vbTab | Chr(9) | Horizontal tab |
| vbVerticalTab | Chr(11) | Vertical tab |

## Tristate Constants

| Constant | Value | Description |
|---|---|---|
| TristateTrue | -1 | True |
| TristateFalse | 0 | False |
| TristateUseDefault | -2 | Use the default setting |

## VarType Constants

| Constant | Value | Description |
|---|---|---|
| vbEmpty | 0 | Uninitialized (default) |
| vbNull | 1 | Contains no valid data |
| vbInteger | 2 | Integer subtype |
| vbLong | 3 | Long subtype |
| vbSingle | 4 | Single subtype |
| vbDouble | 5 | Double subtype |
| vbCurrency | 6 | Currency subtype |

| Constant | Value | Description |
|---|---|---|
| vbDate | 7 | Date subtype |
| vbString | 8 | String subtype |
| vbObject | 9 | Object |
| vbError | 10 | Error subtype |
| vbBoolean | 11 | Boolean subtype |
| vbVariant | 12 | Variant (only used for arrays of variants) |
| vbDataObject | 13 | Data access object |
| vbDecimal | 14 | Decimal subtype |
| vbByte | 17 | Byte subtype |
| vbArray | 8192 | Array |

## Appendix D

# SQL Reference

**In This Appendix:**

- **SQL Statements**
- **SQL Functions**
- **SQL Global Variables**
- **SQL Server Stored Procedures**
- **SQL Server Data Types**

This appendix provides a brief reference for the Microsoft SQL Server statements and stored procedures that you will find yourself using most often while programming Active Server Pages. However, SQL is a complicated language, and this appendix cannot even begin to cover all its nuances. The syntax of many of the SQL statements and functions has been simplified in this appendix. To learn everything you'd ever need to know about using SQL with Microsoft SQL Server, rush to your local bookstore and buy Microsoft SQL Server 7.0 Unleashed by Greg Mable, et al. (1999, Sams Publishing, ISBN 0-672-31227-1).

# SQL Statements

The following SQL statements enable you to create and remove database tables and stored procedures, retrieve data from a database table, modify table data, and grant permissions on database objects.

## SELECT

SELECT [TOP [PERCENT]] *select_list*
FROM *table_source*
WHERE *search_condition*
ORDER BY order_by_expression

The SELECT statement is used to retrieve one or more rows from a database table. Instead of listing particular column names in the *select_list*, you can also use * as a wildcard character to represent all columns.

### Examples

- Using SELECT to retrieve all the columns and all the rows from the Authors table:

  SELECT * FROM Authors

- Using SELECT to retrieve the last name of the author who has the first name Andrew:

  SELECT au_lname
  FROM Authors
  WHERE au_fname = 'Andrew'

- Using SELECT to retrieve a list of publishers and book titles. The results are ordered by the name of the publisher:

  SELECT pub_name, title

  FROM publishers,titles
  WHERE publishers.pub_id = titles.pub_id
  ORDER BY pub_name

- Using SELECT to retrieve the names of the first 10 authors from the Authors table in order of the last name.

  ```
  SELECT TOP 10 au_fname, au_lname
  FROM Authors
  ORDER BY au_lname
  ```

# INSERT

INSERT *table_name* ( *column_list* )
VALUES ( *value_list* )

The INSERT statement is used to insert one or more rows into a database table (Microsoft Access uses INSERT INTO).

## Examples

- Using INSERT to add a new username and password to a Passwords table.

  ```
  INSERT Passwords ( username, password )
  VALUES ( 'Andrew', 'Jones' )
  ```

- Using INSERT with SELECT to insert multiple rows into the newPasswords table from the oldPasswords table:

  ```
  INSERT newPassword ( username, password )
  SELECT username, password
  FROM oldPasswords
  ```

# UPDATE

UPDATE *table_name*
SET *column_name* = *value*
WHERE search_condition

The UPDATE statement is used to update one or more rows in a database table.

## Examples

- Using UPDATE to change the password in every row in a table named Passwords where the username column has the value Andrew Jones:

  ```
  UPDATE Passwords
  SET password = 'secret'
  WHERE username = 'Andrew Jones'
  ```

- Using UPDATE to change the values of multiple columns all at once:

  ```
  UPDATE Passwords
  SET username = 'Bill Gates',
      password = 'Billions'
  WHERE username = 'Andrew Jones'
  ```

# DELETE

DELETE *table_name*
WHERE *search_condition*

The DELETE statement is used to delete one or more rows from a database table.

## Example

- Using DELETE to remove all the rows from the Passwords table in which the password column has the value secret:

  ```
  DELETE Passwords
  WHERE password = 'secret'
  ```

# TRUNCATE TABLE

TRUNCATE TABLE *table_name*

This statement efficiently removes all the rows from a table.

## Example

- Using TRUNCATE TABLE to remove all the rows from the Passwords table:

  TRUNCATE TABLE *Passwords*

# CREATE TABLE

CREATE TABLE *table_name*
( column_name data_type [,...n] )

The CREATE TABLE statement is used to create a new database table. See the later section, "SQL Server Data Types," for a list of data types that you can use when defining the columns for a table.

## Examples

- Using CREATE TABLE to create a table named Passwords that contains usernames and passwords:

  ```
  CREATE TABLE Passwords
  ( username VARCHAR( 50 ), password VARCHAR( 50 ) )
  ```

- Using CREATE TABLE to create a table with an IDENTITY column:

  ```
  CREATE TABLE WebUsers
  ( user_id INT IDENTITY, username VARCHAR( 50 ) )
  ```

- Using CREATE TABLE to create a table with a column with a default value of the current data and time:

```
CREATE TABLE WebUsers
(
  username VARCHAR( 50 ),
  entrydate DATETIME DEFAULT GETDATE()
)
```

# CREATE PROCEDURE

CREATE PROCEDURE *procedure_name*
[ *@parameter_name data_type* [OUTPUT] ]
[,...*n*]
AS
sql_statement [,...n]

The CREATE PROCEDURE statement is used to create a new SQL Server stored procedure.

## Examples

- Using CREATE PROCEDURE to create a new stored procedure that retrieves all the rows from the Passwords table:

```
CREATE PROCEDURE getPasswords
AS
SELECT * FROM Passwords
```

- Using CREATE PROCEDURE to create a new stored procedure that accepts an input parameter and returns an output parameter:

```
CREATE PROCEDURE getUsername
(
  @password VARCHAR( 50 ),
  @username VARCHAR( 50 ) OUTPUT
)
AS
SELECT @username = username
FROM Passwords
WHERE password = @password
```

# GRANT

GRANT ALL | *permission*
ON *table* | *stored_procedure*
TO security_account

The GRANT statement assigns permissions to use a database object to a database user or role.

## Examples

- Using GRANT to give SELECT permission to the database user named WebUser for the table named Passwords:

```
GRANT SELECT ON Passwords TO WebUser
```

- Granting all permissions on a table named Passwords to the public role.

  GRANT ALL ON Passwords TO public

# DROP TABLE

DROP TABLE table_name

This statement permanently removes a database table.

## Example

- Using DROP TABLE to permanently remove a table named Passwords:

  DROP TABLE Passwords

# DROP PROCEDURE

DROP PROCEDURE procedure_name

This statement permanently removes a stored procedure.

## Example

- Using DROP PROCEDURE to permanently remove a stored procedure named getPasswords:

  DROP PROCEDURE getPasswords

# EXECUTE

EXECUTE
[*@return_code* = ] *procedure_name*
[*@parameter* = *value* | *@variable* [OUTPUT] ]
[,...*n*]

The EXECUTE statement is used to run a SQL stored procedure.

## Examples

- Using EXECUTE to execute the getPasswords stored procedure:

  EXECUTE getPasswords

- Using EXECUTE to execute a stored procedure named getValue that returns a return code:

  DECLARE @returnCode INT
  EXECUTE @returnCode = getValue
  SELECT @returnCode

- Using EXECUTE to execute a stored procedure that has both an input parameter and an output parameter:

```
DECLARE @password VARCHAR( 20 )
EXECUTE getPassword 'Andrew Jones', @password OUTPUT
SELECT @password
```

## USE

USE database_name

The USE statement is used to specify the database in which subsequent SQL statements will be executed.

### Example

- In this example, the USE statement switches the database to the master database. The sp_help statement is used to display all the objects in the database.

```
USE Master
sp_help
```

# SQL Functions

All the following functions can be used with the SELECT statement. The majority of these functions can be used to retrieve summary information about the data stored in a table column.

## GETDATE

GETDATE()

This function returns the current date and time.

### Example

- Using GETDATE() to display the current date and time:

```
SELECT GETDATE()
```

## AVG

AVG( column_name )

The AVG function is used to retrieve the average value for a table column.

### Example

- Using AVG to return the average number of times that all users have visited a Web site.

  ```
  SELECT AVG( user_numvisits )
  FROM WebUsers
  ```

## COUNT

COUNT( * | *column_name* )

The COUNT function is used to retrieve either a count of the number of rows in a database table or a count of the number of rows in which a certain column has a value.

### Examples

- Using COUNT to return the number of rows in a table named WebUsers:

  ```
  SELECT COUNT(*)
  FROM WebUsers
  ```

- Using COUNT to return the number of rows in a table named WebUsers where the username column does not have a NULL value:

  ```
  SELECT COUNT( username )
  FROM WebUsers
  ```

## MAX

MAX( column_name )

The MAX function is used to retrieve the maximum value for a table column.

### Example

- Using MAX to return the maximum number of times that any user has visited a Web site:

  ```
  SELECT MAX( user_numvisits )
  FROM WebUsers
  ```

## MIN

MIN( column_name )

The MIN function is used to retrieve the minimum value for a table column.

### Example

- Using MIN to return the minimum number of times that any user has visited a Web site:

  ```
  SELECT MIN( user_numvisits )
  FROM WebUsers
  ```

## SUM

SUM( column_name )

The SUM function is used to add the values of a column.

### Example

- Using SUM to return total number of times that a Web site has been visited:

  ```
  SELECT SUM( user_numvisits )
  FROM WebUsers
  ```

# SQL Global Variables

The following two global variables can be used with the SELECT statement.

## @@IDENTITY

The SQL Server global @@IDENTITY variable contains the value of an IDENTITY column after a row has been inserted.

### Example

- This example assumes that the table named WebUsers has an IDENTITY column named user_id. After a row is inserted into the table, the value of the IDENTITY column for the new row is returned with the @@IDENTITY variable:

  ```
  INSERT WebUsers ( username ) VALUES ( 'Andrew Jones' )
  SELECT @@IDENTITY
  ```

## @@ROWCOUNT

This variable contains a value representing the number of rows that the last statement affected.

### Example

- This example displays the number of rows that were modified by an UPDATE statement using the @@ROWCOUNT variable:

```
UPDATE Passwords
SET username = 'Bill Gates'
WHERE password = 'Billions'
SELECT @@ROWCOUNT
```

# SQL Server Stored Procedures

The following two system stored procedures can be used to retrieve information about the objects in a database.

## sp_help

sp_help [table_name | procedure_name ]

The sp_help system stored procedure is used to display information on database objects. When used without a table or procedure name, it lists the properties of all the objects in the current database.

### Examples

- Using sp_help to display information about the WebUsers table (Returns information including the table columns, indexes and constraints):

```
sp_help WebUsers
```

- Using sp_help to display information about the stored procedure named getAuthors (Returns information including the date and time the procedure was created):

```
sp_help getAuthors
```

## sp_helptext

sp_helptext procedure_name

The sp_helptext system stored procedure can be used to display the SQL statements that constitute a SQL stored procedure.

### Examples

- Using sp_helptext to display the contents of the getAuthors stored procedure:

```
sp_helptext getAuthors
```

- Using sp_helptext to display the contents of the sp_helptext system stored procedure:

```
USE master
sp_helptext sp_helptext
```

# SQL Server Data Types

You can use any of the data types in Tables D.1–D.6 when defining table columns or local variables:

**Table D.1** Character Data Types

| Data Type | Description |
|---|---|
| CHAR | Fixed-length character data with a maximum size of 8,000 characters. |
| VARCHAR | Variable-length character data with a maximum size of 8,000 characters. |
| TEXT | Variable-length character data with a maximum size of 2,147,483,647 characters. |
| NCHAR | Unicode fixed-length character data with a maximum size of 4,000 characters. |
| NVARCHAR | Unicode variable-length character data with a maximum size of 4,000 characters. |
| NTEXT | Unicode Variable-length character data with a maximum size of 1,073,741,823 characters. |

**Table D.2** Numeric Data Types

| Data Type | Description |
|---|---|
| BIT | Has the value 0 or 1. |
| INTEGER | Integer data between –2,147,483,648 and 2,147,483,647. |
| SMALLINT | Integer data between –32,768 and 32,767. |
| NUMERIC | Fixed precision and scale numeric data between –10^38 –1 and 10^38 –1. |
| DECIMAL | Same as NUMERIC |
| FLOAT | Floating precision data type between –1.79E + 308 through 1.79E + 308. |
| REAL | Floating precision data type between –3.40E + 38 through 3.40E + 38. |

**Table D.3** Date and Time Data Types

| Data Type | Description |
|---|---|
| DATETIME | Can have a value between January 1, 1753, and December 31, 9999 (accurate to 3.33 milliseconds) |
| SMALLDATETIME | Can have a value between January 1, 1900, through June 6, 2079 (accurate to one minute) |

**Table D.4** Money Data Types

| Data Type | Description |
|---|---|
| MONEY | Can have a value between –922,337,203,685,477.5808 and 922,337,203,685,477.5807. |
| SMALLMONEY | Can have a value between –214,748.3648 and 214,748.3647. |

**Table D.5** Binary Data Types

| Data Type | Description |
|---|---|
| BINARY | Fixed-length binary data with a maximum size of 8,000 bytes. |
| VARBINARY | Variable-length binary data with a maximum size of 8,000 bytes. |
| IMAGE | Variable-length binary data with a maximum size of 2,147,483,647 bytes. |

**Table D.6** Miscellaneous Data Types

| Data Type | Description |
|---|---|
| CURSOR | Used with stored procedures that have a reference to a cursor as an OUTPUT parameter. |
| TIMESTAMP | Provides a database-wide unique identifier. |
| UNIQUEIDENTIFIER | Provides a Globally Unique Identifier (GUID). |

# Index

## Symbols



## A

## B

## C

## D

## E

## F

## G



## H

## I

## J

## K



## L

**M**

## N

## O

## P

## Q

## R

## S

**T**

## U

## V

## W

## X



## Y-Z